

从校园到职场
CONG XIAOYUAN DAO ZHICHANG

西门子S7－200PLC的使用经验与技巧

第2版

● 肖宝兴　主编





机械工业出版社
CHINA MACHINE PRESS

VISIT…

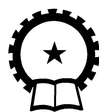
LANZAROTE
Caliente.com

从校园到职场

西门子 S7-200 PLC 的使用经验与技巧

第 2 版

肖宝兴　主编



机 械 工 业 出 版 社

本书从工程应用的角度出发，以德国西门子公司的 S7-200 型 PLC 为样机，突出应用性和实践性，重点介绍了小型 PLC 的硬件结构、工作原理、指令系统、工程应用、程序编辑和程序调试等。通过大量的、有针对性的工程实例，可了解在进行 PLC 控制系统设计时的设计思想、工作步骤、指令运用等，以及一些特殊功能指令的使用方法。书中语言通俗易懂、指令应用丰富、程序简捷全面，以利于读者尽快学习并掌握可编程序控制器技术。

本书可作为刚刚走出校门、初涉电气工程及工业自动化领域的大专院校毕业生的自学用书，亦可作为大专院校相关专业的教材。对于广大的电气工程技术人员也是一本更新知识结构和实践新技术应用的参考书。

图书在版编目（CIP）数据

西门子 S7-200 PLC 的使用经验与技巧 / 肖宝兴主编. —2 版.
—北京：机械工业出版社，2011.3（2014.7 重印）
（从校园到职场）
ISBN 978-7-111-33698-3

Ⅰ. ①西…　Ⅱ. ①肖　Ⅲ. ①可编程序控制器—基本知识
Ⅳ. ①TP332.3

中国版本图书馆 CIP 数据核字（2011）第 040066 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码　100037）
策划编辑：牛新国
责任印制：乔　宇
三河市国英印务有限公司印刷
2014 年 7 月第 2 版第 3 次印刷
169mm×239mm・22 印张・429 千字
标准书号：ISBN 978-7-111-33698-3
定价：48.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
社服务中心：（010）88361066	教材网：http://www.cmpedu.com
销售一部：（010）68326294	机工官网：http://www.cmpbook.com
销售二部：（010）88379649	机工官博：http://weibo.com/cmp1952
读者购书热线：（010）88379203	**封面无防伪标均为盗版**

第 2 版前言

本书的第 1 版于 2008 年 9 月出版，2009 年 10 月出版社找我，让我整理、修改、补充，出第 2 版，因当时给的时间较少，我又十分繁忙，就没有达成。2010 年 4 月出版社又找我，这次让我自己定交稿时间，我愧疚地接受了盛情。

在出第一版时，我们已给本书定位，它属于《从校园到职场》丛书，适合刚刚毕业、走向职场的读者参考。两年多来，我抽空也收集一些信息，感觉大家很会使用本书，这里有本科毕业生以书中的实例为脚本完成毕业设计；有年轻教师参考书中实例在学报上发表文章；有的学生去应聘，把自学了本书也作为一个条件写进材料里；有一次我去劳动技能鉴定所讲 PLC，在上机练习时，我出了书中一例，发现有位学员很快就完成了，我与他交谈，他从包里拿出这本书。种种信息令我感动、令我自豪、令我钦佩。

本书的第一版出版后，我们也不断地发现书中存在一些问题，发现问题就做个记号，这次出版力争都纠正过来，尽管如此，难免还存在问题，恳请广大读者批评指正。因这次出版又增加了几个实例，限于篇幅，只好删除一些内容，几经斟酌，把几个有重复感的、缺少代表性的例子删除了。语言修辞也有些调整。

PLC 已是当今工业控制的主要手段和控制核心，能力在不断加强，好像已成为衡量生产设备自动化控制程度的标志。从事自动化专业的工程技术人员应该掌握这门实用技术，对于这样一门应用性广泛的专业技术，必须通过各种形式的具体实例来感悟它，强化工程意识，提高应用能力。

本书的编写原则是从工程实际出发，由易到难，循序渐进，把一些工程上的典型应用作为学习内容。在理解 PLC 的工作原理，熟悉 PLC 的结构组成及掌握 PLC 的指令系统之后，开始接触工程实例。通过学习，对于提高 PLC 的编程水平和应用能力都会有很大的帮助。

全书共分六章。第一章主要阐述现代工业控制从继电器控制发展到 PLC 控制的过程，简要介绍了 PLC 硬件结构和工作原理。第二章以 S7-200 PLC 为背景讲述了 PLC 系统基本组成、性能特点、基本功能、寻址方式、编程语言、程序结构等。第三章讲解、罗列了 PLC 的基本指令及应用指令，为掌握指令和应用指令打下了基础。第四章是应用实例，由浅到深，由易到难，列举了 20 个应用实例。第五章的系统设计实例相对来说难度大一些，程序更复杂，所涉及的指令功能更强，应用领域更广泛。第六章介绍了 STEP 7-Micro/WIN 编程软件及程序的运行、监控和调试方法。

本书由肖宝兴主编。前言及第五章的例一至例十一由肖宝兴编写；第一章由赵春雷、王伟共同编写；第二章由肖亮编写；第三章由王兆珍编写；第四章由刘兵、肖宝兴共同编写；第五章的例十二、例十三由江彦娥、何善印共同编写；第六章由钱锦锋、吴鹏共同编写。全书由肖宝兴统稿。书中部分内容的编写参照了有关文献，恕不一一列举，在此谨对书后所有参考文献的作者表示感谢。

本书的编写得到了天津职业技术师范大学卢胜利教授、田立国副教授、天津理工大学中环信息学院武波博士的大力支持，在此表示衷心的感谢。

在本书的编写过程中，还得到了潘征、肖璠、吴兴利、王永亮等人的大力支持，在此一同表示感谢。

由于时间仓促，加之作者水平有限，书中错误和不妥之处恳请专家、同仁及广大读者批评指正。

主　编

目　　录

第 2 版前言
第一章　可编程序控制器概述 ……1
第一节　PLC 的产生、定义及分类 ……1
一、PLC 的产生 ……1
二、PLC 的定义 ……3
三、PLC 的分类 ……3
第二节　PLC 的特点、主要功能及性能指标 ……6
一、PLC 的特点 ……6
二、PLC 的主要功能 ……8
三、性能指标 ……9
第三节　PLC 的编程语言 ……10
第四节　PLC 的硬件结构及工作原理 ……14
一、PLC 的硬件结构 ……14
二、PLC 的工作原理 ……19
第五节　PLC 的应用设计及发展方向 ……24
一、PLC 的应用设计 ……24
二、PLC 的发展方向 ……25
第二章　S7-200 PLC 概述 ……28
第一节　S7-200 PLC 的系统组成 ……28
一、S7-200 的系统基本构成 ……28
二、主机单元 ……29
三、数字量扩展模块 ……31
四、模拟量扩展模块 ……31
五、智能模块 ……32
六、其他设备 ……32
第二节　S7-200 PLC 的性能特点及基本功能 ……33
一、S7-200 PLC 的主要技术性能指标 ……33
二、S7-200 PLC 的输入/输出系统 ……34
三、存储系统 ……38
四、S7-200 PLC 的工作方式及扫描周期 ……40

第三节 S7-200 PLC 的内部资源及寻址方式……41
一、S7-200 PLC 的基本数据类型……41
二、软元件（编程元件）……42
三、CPU 存储区域（软元件）的直接寻址……47
四、CPU 存储区域（软元件）的间接寻址……49
五、软元件及操作数的寻址范围……50
第四节 S7-200 PLC 的编程语言和程序结构……52
一、S7-200 PLC 的编程语言……52
二、S7-200 PLC 的程序结构……55
第三章 S7-200 PLC 基本指令与应用指令……58
第一节 位操作指令……58
一、基本逻辑指令……58
二、定时器指令……62
三、计数器指令……66
四、比较指令……69
第二节 数据处理指令……71
一、传送类指令……71
二、移位指令……73
第三节 运算指令……77
一、算术运算指令……78
二、增减指令……80
三、逻辑运算指令……81
第四节 转换指令……83
一、七段显示码指令 SEG……83
二、数据类型转换指令……83
第五节 程序控制指令……84
一、跳转指令……84
二、循环指令……86
三、子程序指令……87
第六节 特殊指令……88
一、中断指令……88
二、高速计数器指令……90
三、通信指令……94
四、PID 回路控制指令……97
第四章 应用实例……100

例一、运料小车延时正、反转控制……100
例二、电动机星形-三角形减压起动控制……103
例三、带式运输机可重复顺序起动、逆序停止控制……105
例四、顺控指令的应用（控制要求同例三）……109
例五、运料小车自动装料、卸料控制……113
例六、四组抢答器程序设计……114
例七、单按钮控制彩灯循环……120
例八、洗衣机自动控制程序设计……122
例九、高速计数器应用程序……128
例十、基于高速计数器的电梯层高自学习及层标显示控制……131
例十一、人行道交通信号灯控制……137
例十二、步进电动机运转控制……143
例十三、两台 PLC 主从式通信……147
例十四、30/5 型桥式起重机小车运行的 PLC 改造程序……153
例十五、外部输入信号中断……157
例十六、利用“定时中断”的彩灯循环左移……159
例十七、Z3050 型摇臂钻床的 PLC 改造……161
例十八、用 TD200 文本显示器监控密码锁开启……166
例十九、变频器控制电动机实现 15 段速运转控制……178
例二十、饮料自动售货机控制程序……183
第五章　系统设计实例……188
例一、百天倒计时控制程序……188
例二、圆形停车库汽车存取控制程序……206
例三、T68 镗床的 PLC 改造程序……213
例四、X62W 万能铣床的 PLC 改造程序……218
例五、两台 PLC 主从式通信程序设计……223
例六、用 TD200 监控邮包配送的程序设计……227
例七、基于 PID 的食品罐头杀菌温度控制程序设计……230
例八、基于 USS 协议库的 PLC 与变频器的通信……234
例九、PLC 改造 B2012A 型龙门刨床控制系统……244
例十、波浪式喷泉的 PLC 控制程序……264
例十一、变频器调速系统远程监控的实现……268
例十二、车间生产流水线产品运输控制程序……288
例十三、机械手搬运产品控制程序……295
第六章　STEP 7-Micro/WIN 编程软件……303

第一节　软件安装和设置 ······ 303
第二节　STEP 7-Micro/WIN 简介 ······ 308
第三节　定制 STEP7-Micro/WIN ······ 313
第四节　编程计算机与 CPU 通信 ······ 315
第五节　程序的编写与传送 ······ 320
第六节　程序的运行监控与调试 ······ 333
第七节　通信程序下载与向导编程 ······ 340
参考文献 ······ 344

第一章　可编程序控制器概述

可编程序控制器（Programmable Logic Controller，简称 PLC），它是以微处理器为核心的通用工业控制装置，是在继电器-接触器控制基础上发展起来的。随着现代社会生产的发展和技术进步，现代工业生产自动化水平的日益提高及微电子技术的迅猛发展，当今的 PLC 已将 3C（Computer、Control、Communication）技术，即微型计算机技术、控制技术及通信技术融为一体，在控制系统中又能起到“3 电”控制作用，即电控、电仪、电信这三个不同作用的一种高可靠性控制器，是当代工业生产自动化的重要支柱。

第一节　PLC 的产生、定义及分类

一、PLC 的产生

PLC 产生以前，以各种继电器为主要元件的电气控制线路承担着生产过程自动控制的艰巨任务。这些器件组成的控制系统需要大量的导线，大量的控制柜，占据大量的空间。当这些继电器运行时又产生大量的噪声，消耗大量的电能。为保证控制系统正常运行，需要安排大量的电气技术人员进行维护，有时某个继电器的损坏，甚至某个继电器的触点接触不良都会影响整个系统的正常运行。检查和排除故障又是非常困难的，现场电气技术人员的技术水平也直接影响设备恢复运行的速度。尤其是在生产工艺发生变化时，可能需要增加很多继电器或继电器控制柜，重新接线或改线的工作量极大，甚至可能需要重新设计控制系统。面对这种局面，人们迫切需要一种新的工业控制装置来取代传统的继电器控制系统，使电气控制系统工作更可靠、更容易维修、更能适应经常变化的生产工艺的要求。

到 20 世纪 60 年代，由于小型计算机的出现和大规模生产及多机群控的发展，人们曾试图用小型计算机来实现工业控制的要求，但由于价格高、输入/输出电路不匹配和编程技术复杂等原因，一直未能得到推广应用。

20 世纪 60 年代末期，美国的汽车制造业竞争激烈。各生产厂家的汽车型号不断更新，它必然要求生产线的控制亦随之改变，以及对整个控制系统重新配置。为此要寻求一种比继电器更可靠、响应速度更快、功能更强大的通用工业控制器。GM 公司提出了著名的 10 条技术指标在社会上招标，要求控制设备制造商为其生

产线提供一种新型的通用工业控制器，它应具有以下特点：

① 编程简单，可在现场修改程序；

② 维修方便，采用插件式结构；

③ 可靠性高于继电器控制装置；

④ 体积小于继电器控制盘；

⑤ 数据可直接进入管理计算机；

⑥ 成本可与继电器控制盘竞争；

⑦ 输入可以是交流 115V（美国电压标准）；

⑧ 输出为交流 115V，2A 以上；

⑨ 扩展时原系统改变最小；

⑩ 用户存储器至少能扩展到 4KB。

1969 年美国数据设备公司（DEC）根据上述要求，研制开发出世界上第一台可编程序控制器，并在 GM 公司汽车生产线上首次应用成功，取得了显著的经济效益。当时人们把它称为可编程序逻辑控制器（Programmable Logic Controller），简称 PLC。

可编程序控制器这一新技术的出现，受到国内外工程技术界的极大关注。纷纷投入力量研制。第一个把 PLC 商品化的是美国哥德电子公司（GOULD Inc），时间也是 1969 年。1971 年，日本从美国引进了这项新技术，研制出日本第一台可编程序控制器。1973～1974 年，德国和法国也都相继研制出自己的可编程序控制器，德国西门子公司（SIEMENS）于 1973 年研制出欧洲第一台 PLC。我国从 1974 年开始研制，1977 年开始工业应用。

20 世纪 70 年代后期，随着微电子技术和计算机技术的发展，可编程序逻辑控制器具备更多的计算机功能，不仅用逻辑编程取代硬接线逻辑，还增加了运算、数据传送和处理等功能，真正成为一种电子计算机工业控制装置，而且做到了小型化和超小型化。这种采用微电脑技术的工业控制装置的功能远远超出逻辑控制、顺序控制的范围，故称为可编程序控制器，简称 PC（Programmable Controller）。但由于 PC 容易和个人计算机（Personal Computer）混淆，故人们仍习惯地用 PLC 作为可编程序控制器的缩写。

进入 20 世纪 80 年代以来，随着大规模和超大规模集成电路等微电子技术的迅猛发展，以 16 位和 32 位微处理器构成的微机化 PLC 得到了惊人的发展，使 PLC 在概念、设计、性能价格比以及应用等方面都有了新的突破，不仅控制功能增强，功耗、体积减小，成本下降，可靠性提高，编程和故障检测更为灵活方便，而且远程 I/O 和通信网络、数据处理以及图像显示也有了长足的发展。所有这些已经使 PLC 应用于连续生产的过程控制系统，使之成为现代工业生产自动化的四大支柱之一。

二、PLC 的定义

PLC 一直在飞速发展中，因此到现在为止，还未能对其下一个十分确切的定义。

1980 年，美国电器制造商协会（National Electronic Manufacture Association，NEMA）将可编程序控制器定义为：“可编程序控制器是一种带有指令存储器，数字的或模拟的输入/输出接口，以位运算为主，能完成逻辑、顺序、定时、计数和算术运算等功能，用于控制机器或生产过程的自动控制装置”。

1982 年 11 月国际电工委员会（IEC）曾颁发了可编程序控制器标准草案第一稿，1985 年 1 月发表了第二稿，1987 年 2 月颁布了第三稿。该草案中对可编程序控制器的定义是：“可编程序控制器是一种数字运算操作的电子系统，专为工业环境而设计。它采用了可编程序的存储器，用来在其内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令，并通过数字式或模拟式的输入和输出，控制各种类型机械的生产过程。而有关的外围设备，都应按易于与工业系统连成一个整体，易于扩充其功能的原则设计”。

定义强调了 PLC 应直接应用于工业环境，它必须具有很强的抗干扰能力、广泛的适应能力和应用范围。这也是区别于一般微机控制系统的一个重要特征。

定义强调了 PLC 是“数字运算操作的电子系统”，它也是一种计算机。它是“专为工业环境下应用而设计的”工业计算机。这种工业计算机采用“面向用户的指令”，因此编程方便，它能完成逻辑运算、顺序控制、定时、计数运算等操作，它还具有数字量和模拟量输入和输出的能力，并且非常容易与工业控制连成一体，易于扩充。

应该强调的是，PLC 与以往所讲的鼓式、机械式的顺序控制器在“可编程”方面有质的区别。由于 PLC 引入了微处理器及半导体存储器等新一代电子器件，并用规定的指令进行编程，可以灵活地修改程序，即它是用软件方式来实现“可编程”的目的。

三、PLC 的分类

PLC 发展到今天，已经有很多种形式，而且功能也不尽相同，分类时，一般按以下原则来考虑。

1. 根据控制规模分类

PLC 的控制规模是以所配置的输入/输出点数来衡量的，PLC 的 I/O 点数表明了 PLC 可从外部接收多少个输入信号和向外部发出多少个输出信号，实际上也就是 PLC 的输入/输出端子数。根据 I/O 点数的多少可将 PLC 分为小型机、中型机和大型机，一般来说，点数多的 PLC，功能也相应较强。

（1）小型机　小型 PLC 的功能一般以开关量控制为主，小型 PLC 输入/输出总点数一般在 256 点以下，用户程序存储器容量在 4K 字左右。现在的高性能小型 PLC 还具有一定的通信能力和少量的模拟量处理能力，这类 PLC 的特点是价格低廉、体积小巧，适合于控制单台设备和开发机电一体化产品。

典型的小型机有欧姆龙公司的 C 系列，三菱公司的 F1 系列，西门子公司的 S5-100U，S7-200 系列等。

（2）中型机　I/O 总点数在 256～1024 之间的称为中型机，它除了具备逻辑运算功能，还增加了模拟量输入/输出、算术运算、数据传送、数据通信等功能，可完成既有开关量又有模拟量的复杂控制。用户程序存储器容量达到 8K 字左右。中型机的软件比小型机丰富，在已固化的程序内，一般还有 PID（比例、积分、微分）调节，整数/浮点运算等功能模板。

中型机的特点是功能强，配置灵活，适用于具有诸如温度、压力、流量、速度、角度、位置等模拟量控制和大量开关量控制的复杂机械，以及连续生产过程控制场合。

（3）大型机　I/O 总点数在 1024 点以上的称为大型机，用户程序存储器容量达到 16K 字以上，大型 PLC 的功能更加完善，具有数据运算、模拟调节、联网通信、监视记录、打印等功能。大型机的内存容量超过 640KB，监控系统采用 CRT 显示，能够表示生产过程的工艺流程，记录各种曲线，PID 调节参数选择图等，能进行中断控制、智能控制、远程控制等。

大型机的特点是 I/O 点数特别多，控制规模宏大，组网能力强。可用于大规模的过程控制，构成分布式控制系统，或者整个工厂的集散控制系统。

典型的 PLC 大型机有西门子公司的 S7-400，欧姆龙公司的 CVM1 和 CS1 系列，AB 公司的 SLC5/05 等系列产品。以上划分没有十分严格的界限，随着 PLC 技术的飞速发展，某些小型 PLC 也具有中型或大型 PLC 的功能，这也是 PLC 的发展趋势。

2. 根据结构形式分类

根据 PLC 结构形式的不同，可分为整体式、模板式及分散式 3 种形式。

（1）整体式　这种结构的特点是将 PLC 的基本部件，如 CPU 板、输入板、输出板、电源板等都集中配置在一个箱体中，安装在一个标准机壳内，构成一个整体，有的甚至全部装在一块印制电路板上，组成 PLC 的一个基本单元（主机）或扩展单元。基本单元上设有扩展端口，通过扩展电缆与扩展单元相连，配有许多专用的特殊功能模块，如模拟量 I/O 模块、热电偶、热电阻模块、通信模块等，以构成 PLC 不同的配置。

整体式 PLC 结构紧凑、体积小、重量轻、价格低、容易装配在工业控制设备的内部，比较适合于生产机械的单机控制。

这种结构的缺点是主机的 I/O 点数固定，使用不够灵活，维修也较麻烦。

微型和小型 PLC 一般为整体式结构，如西门子的 S7-200 系列。

（2）模板式　这种结构的 PLC 各部分以单独的模板分开设置。如电源模板、CPU 模板、I/O 模板、各种功能模板及通信模板等。这种 PLC 一般设有机架底板（也有的 PLC 为串行连接，没有底板），在底板上有若干插座，使用时，各种模板直接插入机架底板即可。各模块功能是独立的，外形尺寸是统一的，可根据需要灵活配置，装备方便、维修简单、易于扩展，一般中、大型 PLC 多采用这种结构形式，如西门子的 S7-300 和 S7-400 系列。

这种结构形式的缺点是结构较复杂，各种插件多，因而增加了造价。

（3）分散式　所谓分散式的结构就是将 PLC 的电源、CPU、存储器集中放置在控制室，而将各 I/O 模板分散放置在各个工作站，由通信接口进行通信连接，由 CPU 集中指挥。

以上 3 种形式的可编程序控制器的外观结构示意如图 1-1 所示。

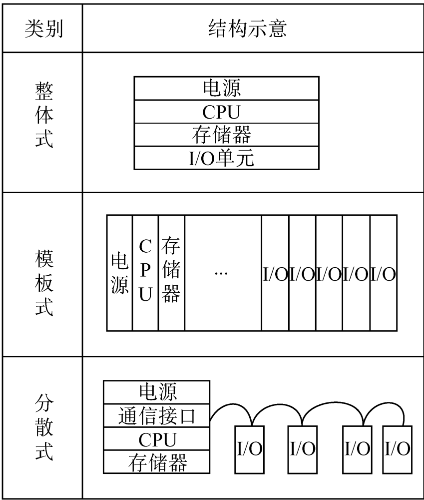


图 1-1　可编程序控制器的外观结构示意图

3. 根据用途分类

（1）用于顺序逻辑控制　顺序逻辑控制是可编程序控制器的最基本的控制功能，也是 PLC 应用最多的场合，比较典型的应用如自动电梯的控制，自动仓库的自动存取，各种管道上的电磁阀的自动开启和关闭，带式运输机的顺序起动，或者自动化生产线的多机控制等，这些都是顺序逻辑控制。要完成这类控制，不要求 PLC 有太多的功能，只要有足够数量的 I/O 回路即可，因此可选低档的 PLC。

（2）用于闭环过程控制　对于闭环控制系统，除了要用开关量 I/O 实现顺序逻辑控制外，还要有模拟量的 I/O 回路，以供采样输入和调节输出，实现过程控

制中的 PID 调节，形成闭环过程控制系统，而中期的 PLC 由于具有数值运算和处理模拟量信号的功能，可以设计出各种 PID 控制器。随着 PLC 控制规模的增大，可控制的回路数已从几个增加到几十个甚至几百个，因此可实现比较复杂的闭环控制系统，实现对温度、压力、速度等物理量的连续调节。比较典型的应用如加热炉的温度，锅炉的自动给水控制等。要完成这类控制，不仅要求 PLC 有足够数量的 I/O 点，还要有模拟量的处理能力，因此对 PLC 的功能要求高。根据能处理的模拟量的多少，至少应选用中档 PLC。

（3）用于多级分布式和集散控制系统　对于这样档次的控制要求，除了要求所选用的 PLC 具有上述的功能外，还要求具有较强的通信功能，以实现各工作站之间的通信，上位机与下位机的通信，最终实现全厂的自动化，形成通信网络。由于近期推出的 PLC 都具有很强的通信和联网功能，建立一个自动化工厂已成为可能。显然，近期推出的就是档次最高的。

4. 根据生产厂家分类

PLC 的生产厂家众多，各厂家的 PLC，其点数、容量、功能各有差异，但都自成系列，指令及外设向上兼容，因此在选择 PLC 时，若选择同一系列的产品，则可以使系统构成容易，操作人员使用方便，备品配件的通用性及兼容性好。比较有代表性的有：日本欧姆龙公司的 C 系列，三菱公司的 F 系列，美国 AB 公司的 PLC-5 系列，德国西门子公司的 S5 系列、S7 系列等。

第二节　PLC 的特点、主要功能及性能指标

一、PLC 的特点

现代工业生产过程是多种多样的，它们对控制的要求也各不相同，为了能够在各种工业环境中使用 PLC，所有生产厂家的 PLC 都有许多共同的特点。

1. 抗干扰能力强，可靠性极高

工业生产对电气控制设备的可靠性的要求是非常高的，它应具有很强的抗干扰能力，能在很恶劣的环境下（如温度高、湿度大、金属粉尘多、距离高压设备近、有较强的高频电磁干扰等）长期连续可靠地工作，平均无故障时间长，故障修复时间短。而 PLC 是专为工业环境设计的，它在电子线路、机械结构以及软件结构上都吸取了生产厂家长期积累的生产控制经验，主要模块均采用大规模与超大规模集成电路，I/O 系统设计有完善的通道保护与信号调理电路，在结构上对耐热、防潮、防尘、抗震等都有周到的考虑；在硬件上采用隔离、屏蔽、滤波、接地等抗干扰措施；在软件上采用数字滤波等抗干扰和故障诊断措施，所有这些使 PLC 具有较高的抗干扰能力。PLC 的平均无故障时间通常在几万小时甚至几十万小时以上，这是其他电气控制设备根本做不到的。

另外，PLC 特有的循环扫描的工作方式，有效地屏蔽了绝大多数的干扰信号。通过这些有效的措施，保证了可编程序控制器的高可靠性。

2. 编程方便

PLC 是面向工业企业中一般电气工程技术人员而设计的，设计者充分考虑到现场工作人员的技能和习惯，采用易于理解和掌握的梯形图语言，以及面向工业控制的简单指令。这种梯形图语言既继承了传统继电器控制线路的表达形式（如线圈、触点、动合、动断），又考虑到工业企业中的电气技术人员的看图习惯和微机应用水平。因此，梯形图语言对于企业中熟悉继电器控制线路的电气工程技术人员是非常亲切的。它形象、直观、简单、易学，尤其是对于小型 PLC 而言，几乎不需要专门的计算机知识，只要进行短暂几天甚至几小时的培训，就能基本掌握编程方法。立足于这样的出发点，经过几十年的验证，它真正受到了广大电气技术人员的欢迎。

3. 使用方便

PLC 及其扩展模块品种繁多，所构成的产品已系列化和模块化，并且配有品种齐全的各种软件，用户可灵活组合成各种大小和不同要求的控制系统。在由 PLC 组成的控制系统中，我们只需要在 PLC 的输入/输出端子上接入相应的导线即可。而导线的另一端可以接按钮、限位开关、继电器线圈、接触器线圈等，大量而又繁杂的中间环节的硬接线线路不见了。在生产工艺流程改变或生产线设备更新、或系统控制要求改变，需要变更控制系统的功能时，除了 I/O 通道上的外部接线需做很小的调整外，只是把用户程序做相应的修改就可以了。同一个 PLC 装置用于不同的控制对象，只是输入/输出的组件和应用软件不同。PLC 的输入/输出可直接与交流 220V，直流 24V 等相连，并有较强的带负载能力。

4. 维护方便

用户所编写的控制程序可通过编程器输入到 PLC 的存储器中。当 PLC 工作时编程器还可随时监控，使得 PLC 的操作及维护都很方便。PLC 还具有很强的自诊断能力，能随时检查出自身的故障，并显示给操作人员，如 I/O 通道的状态，RAM 的后备电池的状态，数据通信的异常，PLC 内部电路的异常等信息。正是通过 PLC 的这种完善的诊断和显示能力，当 PLC 主机或外部的输入装置及执行机构发生故障时，使操作人员能迅速检查、判断故障原因、确定故障位置，以便采取迅速有效的措施。如果是 PLC 本身故障，在维修时只需要更换插入式模板或其他易损件即可完成，既方便又减少了影响生产的时间。

5. 设计、施工、调试的周期短

用继电器控制完成一项控制工程，必须首先按工艺要求画出电气原理图，然后画出继电器屏（柜）的布置和接线图等，进行安装调试，以后修改起来非常不便。而采用 PLC 控制，由于其硬件、软件齐全，设计和施工可同时进行。用软件

编程取代了继电器硬接线，使得控制柜的设计及安装接线工作量大为减少，具体的程序编制工作也可在 PLC 到货之前进行，因而缩短了设计周期。因为 PLC 是通过程序完成控制任务的，采用了方便用户的工业编程语言，用户程序大都可以在实验室模拟调试，模拟调试好后再进行生产现场联机统调，使得调试方便、快速、安全，因此大大缩短了设计和投运周期。

二、PLC 的主要功能

PLC 是采用微电子技术来完成各种控制功能的自动化设备，可以在现场的输入信号作用下，按照预先输入的程序，控制现场的执行机构按照一定规律进行动作。其主要功能如下：

1. 顺序逻辑控制

这是 PLC 最基本最广泛的应用领域，用来取代继电器控制系统，实现逻辑控制和顺序控制。它既可用于单机或多机控制，又可用于自动化生产线的控制。PLC 根据控制要求准确无误地处理输入信号、输出信号的各种逻辑关系。

2. 运动控制

在机械加工行业，PLC 与计算机数控（CNC）集成在一起，用以完成机床的运动控制，PLC 制造商已提供了拖动步进电动机或伺服电动机的单轴或多轴的位置控制模板，在多数情况下，PLC 把描述目标位置的数据送给模板，模板移动一轴或数轴到目标位置。当每个轴移动时，位置控制模板保持适当的速度和加速度，确保运动平滑。

3. 定时控制

PLC 为用户提供了一定数量的定时器，并设置了定时器指令，为了保证定时精度，定时器的时基单位又分为几个档次，有 0.1s 级、0.01s 级、0.001s 级，也可以按照一定方式进行定时时间的扩展。PLC 定时精度高，定时设定方便、灵活。

4. 计数控制

PLC 为用户提供的计数器分为普通计数器、可逆计数器（增减计数器）、高速计数器等，用来完成不同用途的计数控制。当计数器的当前计数值等于计数器的设定值，或在某一数值范围时，发出控制命令。计数器的计数值可在运行中被读出，也可以在运行中进行修改。

5. 步进控制

PLC 为用户提供了一定数量的移位寄存器，用移位寄存器可方便地完成步进控制功能，在一道工序完成之后，自动进行下一道工序，一个工作周期结束以后，自动进行下一个周期。有些 PLC 还专门设有步进控制指令，使得步进控制更为方便。

6. 数据处理

大部分 PLC 都具有不同程度的数据处理功能，主要可以完成的数据运算如：

加、减、乘、除、乘方、开方等。逻辑运算如：字与、字或、字异或、求反、移位、数据比较和传送及数值的转换等操作。

7. 模/数和数/模转换

在过程控制或闭环控制系统中，存在温度、压力、速度、电流、电压等连续变化的物理量（或称模拟量）。过去，由于 PLC 主要完成逻辑运算控制，对于这些模拟量的控制主要靠仪表控制或分布式控制系统（DCS）。目前，不仅大、中型 PLC 都具有模拟量处理功能，甚至很多小型 PLC 也具有模拟量处理功能，而且编程和使用都很方便。

8. 通信及联网

目前，绝大多数 PLC 都具备了通信能力，能够实现 PLC 与计算机、PLC 与 PLC 之间的通信。通过这些通信技术使 PLC 更容易构成工厂自动化（FA）系统。PLC 也可与打印机、监视器等外部设备相连，记录和监视有关数据。

三、性能指标

性能指标是用户评价和选购机型的依据。目前，市场销售的 PLC 和我国工业企业中使用的 PLC 绝大多数是国外生产的产品。各种机型种类繁多，各个厂家在说明其性能指标时，侧重点也不完全相同。如何评价一台 PLC 的档次高低、规模大小、使用场所，至今还没有一个统一的衡量标准。但是，当用户在进行 PLC 的选型时，可以参照生产厂商提供的技术指标，从以下几个方面考虑：

1. CPU 技术指标

CPU 技术指标是 PLC 各项性能指标中最重要的。在这部分技术指标中，应反映出 CPU 的类型、用户程序存储器容量、可连接的 I/O 接口总点数（开关量多少点，模拟量多少路）、指令长度、指令条数、扫描速度（ms/千字）。

2. I/O 模板技术指标

对于开关量输入模板，要反映出输入点数/块、电源类型、工作电压等级、com 端、输入电路等情况。

对于开关量输出模板，要反映出输出点数/块、电源类型、工作电压等级、com 端、输出电路等情况。一般 PLC 的输出形式有 3 种：继电器输出、晶体管输出、双向晶闸管输出，要根据不同的负载性质选择 PLC 输出电路的形式。

3. 编程器及编程软件

反映这部分性能指标有编程器的形式（简易编程器、图形编程器、通用计算机），运行环境（DOS 或 Windows），编程软件及是否支持高级语言等。

4. 通信功能

随着 PLC 控制功能的不断增强和控制规模的不断增大，使得通信和联网的能力成为衡量现代 PLC 的重要指标。与之有关的是通信接口、通信模块、通信协议

及通信指令等。PLC 的通信可分为两类：一类是通过专用的通信设备和通信协议，在同一生产厂家的各个 PLC 之间进行的通信，另一类是通过通用的通信口和通信协议，在 PLC 与上位计算机或其他智能设备之间进行的通信。

5. 扩展性

PLC 的扩展是指 PLC 的主机配置扩展模板的能力，它体现在两个方面，一个是 I/O（数字量 I/O 或模拟量 I/O）的扩展能力，用于扩展系统的输入/输出点数；另一个是 CPU 模板的扩展能力，用于扩展各种智能模板，如温度控制模板、高速计数器模板、闭环控制模板等，实现多个 CPU 的协调和信息交换。

第三节　PLC 的编程语言

PLC 为用户提供了完整的编程语言，以适应程序用户编制的需要。PLC 提供的编程语言通常有以下几种：梯形图、语句表、功能图和功能块图。

1. 梯形图（LAD）

梯形图（LADDER）是一种图形编程语言，它是从继电器控制原理图的基础上演变而来的。PLC 的梯形图与继电器控制系统原理图的基本思想是一致的，它沿用继电器的触点（触点在梯形图中又常称为接点）、线圈、串并联等术语和图形符号，同时还增加了一些继电器-接触器控制系统中没有的特殊功能符号。对于熟悉继电器控制线路的电气技术人员来说，很容易被接受，且不需要学习专门的计算机知识。因此，在 PLC 应用中，梯形图是最基本的、最普遍的编程语言。需要说明的是，这种编程方式只能用编程软件通过计算机下载到 PLC 当中去。如果使用编程器编程，还需要将梯形图转变为语句表用助记符将程序输入 PLC 中。

PLC 的梯形图虽然是从继电器控制线路图发展而来的，但与其又有一些本质的区别。

（1）PLC 的梯形图中的某些元件沿用了“继电器”这一名称，例如：输入继电器、输出继电器、中间继电器等。但是，这些继电器并不是实际存在的物理继电器，而是“软继电器”，也可以说是存储器。它们中的每一个都与 PLC 的用户程序存储器中的数据存储区中的元件映像寄存器的一个具体存储单元相对应。如果某个存储单元为“1”状态，则表示与这个存储单元相对应的那个继电器的“线圈得电”。反之，如果某个存储单元为“0”状态，则表示与这个存储单元相对应的那个继电器的“线圈断电”。这样，我们就能根据数据存储区中某个存储单元的状态是“1”还是“0”判断与之对应的那个继电器的线圈是否“得电”。

（2）PLC 梯形图中仍然保留了动合触点和动断触点的名称，这些触点的接通或断开，取决于其线圈是否得电（这是继电器、接触器的最基本的工作原理）。在梯形图中，当程序扫描到某个继电器的触点时，就去检查其线圈是否“得电”，即

去检查与之对应的那个存储单元的状态是“1”还是“0”。如果该触点是动合触点，就取它的原状态，如果该触点是动断触点，就取它的反状态。

（3）PLC 梯形图中的各种继电器触点的串并联连接，实质上是将这些基本单元的状态依次取出来，进行“逻辑与”、“逻辑或”等逻辑运算。而计算机对进行这些逻辑运算的次数是没有限制的，因此，可在编制程序时无限次使用这些触点。就像马路上路灯的状态可以被无数眼睛看到那样。特别需要注意的是在梯形图程序中同一个继电器的线圈一般只能使用一次，其触点形式及使用次数是随意的。

（4）如图 1-2 是典型的梯形示意图，左右两条垂直的线称为母线。母线之间是触点的逻辑连线和线圈的输出，不过多数 PLC 现在只保留左母线了。

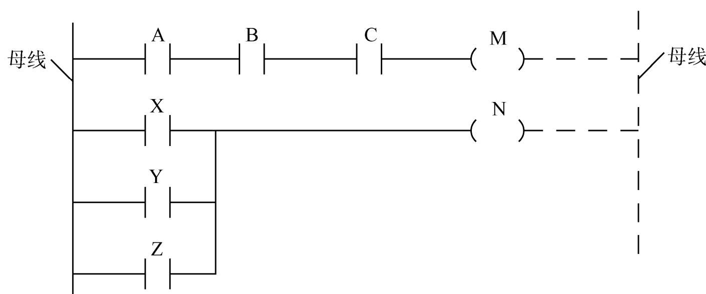


图 1-2 梯形图举例

梯形图的一个关键概念是“能流”（POWER FLOW），这只是概念上的“能流”。图 1-2 中，把左边的母线假想为电源中的“零线”。如果有“能流”从左至右流向线圈，则线圈被激励。“能流”可以通过被激励（ON）的常开触点和未被激励（OFF）的常闭触点自左向右流，如图 1-2 中，当 A、B、C 接点都接通后，线圈 M 才能接通（被激励），只要其中一个触点不接通，线圈就不会接通；而 X、Y、Z 触点中任何一个接通，线圈 N 就被激励。

要强调指出的是，引入“能流”的概念，仅仅是为了和继电器-接触器控制系统相比较，使我们对梯形图有一个深入的认识，其实“能流”在梯形图中是不存在的。

在梯形图中，触点代表逻辑“输入”条件，如开关、按钮和内部条件等；线圈通常代表逻辑“输出”结果，它可驱动像接触器线圈、电磁阀、灯及警铃等直流或单相交流负载。

（5）在继电器控制线路中，各个并联电路是同时加电压，并行工作的，由于实际元件动作的机械惯性可能会发生触点竞争现象。在梯形图中，各个编程元件的动作顺序是按扫描顺序依次执行的，或者说是按串行的方式工作的，在执行梯形图程序时，是自上而下，从左到右，串行扫描，不会发生触点竞争现象。

下面举两个例子说明，表面上看起来完全一样的继电器控制线路图与梯形图，

它们产生的效果可能不完全一样，甚至某些作用完全相反。图 1-3 及图 1-4 给出了两组结构上完全一样的继电器控制线路图与梯形图，但最后的控制结果却不相同，我们先来看图 1-3 的情况。图 1-3 中 a 是继电器控制线路图，图 1-3b 是梯形图。

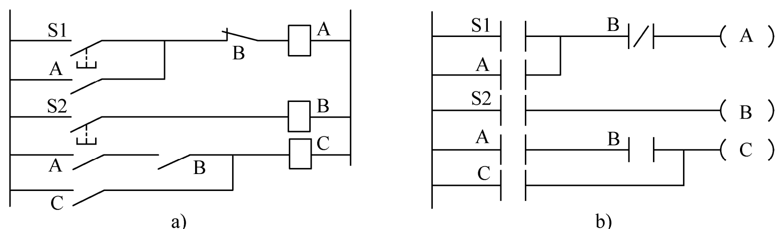


图 1-3　继电器控制线路图不可能实现但梯形图能实现的情况

在图 1-3a 中，当 S1 工作后，A 得电并自保持，且为 C 得电创造条件。接着 S2 动作，使 B 得电。B 的动断点先切断 A，A 的动合点随之断开，此时虽然 B 的动合点闭合，但 A 已断开，使 C 总不能得电，更不用说自保持了。我们再来看看图 1-3b，当 S1 动作后 A“得电”并自保持，在 S2 动作后，B“得电”，本扫描周期内不会再改变 A，当程序扫描到下面的 A、B 动合触点时，因其线圈此时均已“得电”，它们均处于接通状态。这样，C 能“得电”且自保持。待到下个扫描周期时，虽然 A 被复位，动合点断开，但因 C 已自保持，所以 C 不会受到影响。始终处在闭合状态，达到了控制目的。

下面我们再来看图 1-4 的情况，这是个继电器控制线路图能实现但梯形图不能实现的例子，电动机单向连续与点动运行的控制线路，我们先看图 1-4a，当按下 SB1 时，KM 线圈得电，它的动合点随之闭合自锁。实现电动机连续运行；当按下 SB2 时，它的动断点使 KM 的自保持线路断开，实现电动机点动运行。再看图 1-4b，当按下 SB1 时，同样能形成连续运行状态。当按下 SB2 时，就形成不了点动运行状态。具体分析一下，当按下 SB2 时，它的常开触点使 KM“得电”，常闭触点断开了 KM 的自保持线路，当松开 SB2 时，由于 PLC 的周期性逐行扫描的特点，就会形成 SB2 的常开触点是断开了，但常闭触点闭合，而此时此刻的 KM 常开触点取上一个周期 KM 线圈的状态，也即闭合状态。这样一来就仍能形成 KM“得电”，使 KM 不能随着 SB2 的断开而断开，从而也就形成不了点动的功能。

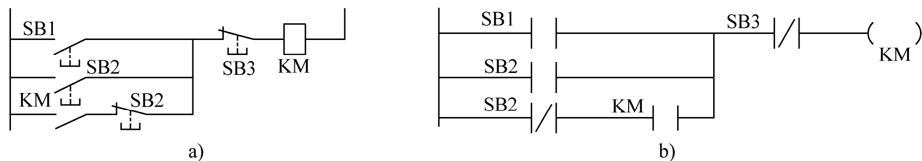


图 1-4　继电器控制线路图能实现但梯形图不可能实现的情况

梯形图语言简单明了，易于理解，是所有编程语言的首选。

2. 语句表（STL）

语句表（Statements List）就是用助记符来表达 PLC 的各种功能，类似于计算机的汇编语言，但比汇编语言通俗易懂，它是 PLC 最基础的编程语言。所谓语句表编程，是用一个或几个容易记忆的字符来代表 PLC 的某种操作功能。这种编程语言可使用简易编程器编程，尤其是在未开发计算机软件时，就只能将已编好的梯形图程序转换成语句表的形式，再通过简易编程器将用户程序逐条输入到 PLC 的存储器中进行编程。通常每条指令由地址、操作码（指令）和操作数（数据或器件编号）三部分组成。语句表编程设备简单，逻辑紧凑，系统化，连接范围不受限制，但比较抽象，一般与梯形图语言配合使用，互为补充。目前，大多数 PLC 都有语句表编程功能。

3. 顺序功能图（SFC）

顺序功能图（Sequence Function Chart）编程方式采用画工艺流程图的方法编程，亦称功能图，只要在每一个工艺方框的输入和输出端标上特定的符号即可。对于在工厂中搞工艺设计的人来说，用这种方法编程，不需要很多的电气知识，非常方便。

不少 PLC 的新产品采用了顺序功能图，提供了用于 SFC 编程的指令，有的公司已生产出系列的、可供不同的 PLC 使用的 SFC 编程器，原来十几页的梯形图程序，SFC 只用一页就可以完成。另外，由于这种编程语言最适合从事工艺设计的工程技术人员，因此，它是一种效果显著、深受欢迎、前途光明的编程语言。目前国际电工委员会（IEC）也正在实施并发展这种语言的编程标准。

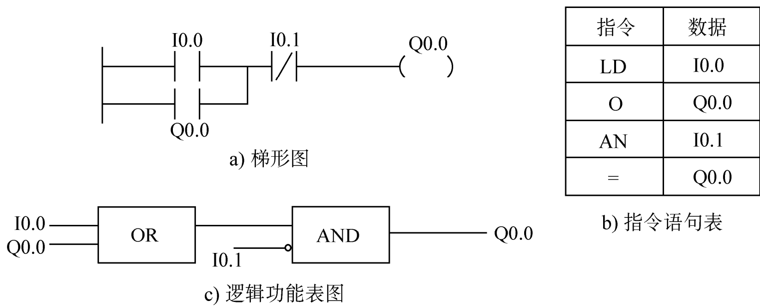
4. 功能块图（FBD）

这是一种由逻辑功能符号组成的功能块图（Function Block Diagrams）来表达命令的编程语言，这种编程语言基本上沿用半导体逻辑电路的逻辑框图。对每一种功能都使用一个运算方块，其运算功能由方块内的符号确定。常用“与”、“或”、“非”等逻辑功能表达控制逻辑。和功能方块有关的输入画在方块的左边，输出画在方块的右边。利用 FBD 可以查到像普通逻辑门图形的逻辑盒指令。它没有梯形图编程器中的触点和线圈，但有与之等价的指令，这些指令是作为盒指令出现的。程序逻辑由这些盒指令之间的连接决定。采用这种编程语言，不仅能简单明确地表达逻辑功能，还能通过对各种功能块的组合，实现加法、乘法、比较等高级功能，所以，它也是一种功能较强的图形编程语言。对于熟悉逻辑电路和具有逻辑代数基础的人来说，是非常方便的。图 1-5 为实现电动机起、停、保控制的 3 种编程语言的表达方式。

5. 高级语言

在一些大型 PLC 中，为了完成一些较为复杂的控制，采用功能很强的微处理

器和大容量存储器，将逻辑控制、模拟控制、数值计算与通信功能结合在一起，配备 BASIC、Pascal、C 等计算机语言，从而可像使用通用计算机那样进行结构化编程，使 PLC 具有更强的功能。



指令	数据
LD	I0.0
O	Q0.0
AN	I0.1
=	Q0.0

b) 指令语句表

图 1-5　3 种编程语言举例

目前，各种类型的 PLC 基本上都同时具备两种以上的编程语言。其中，以同时使用梯形图和语句表的占大多数。而梯形图与语句表在表达方式上，不同厂家、不同型号的 PLC 都还有些差异，使用符号也不尽相同，配置的功能也各有千秋。因此，各个厂家不同系列，不同型号的 PLC 是互不兼容的，但基本的逻辑思想与编程原理是一致的。

第四节　PLC 的硬件结构及工作原理

一、PLC 的硬件结构

从 PLC 的定义我们可知，PLC 也是一种计算机，它有着与通用计算机相类似的结构，即 PLC 也是由中央处理器（CPU），存储器（MEMORY），输入/输出（I/O）接口及电源组成的，现以小型可编程序控制器为例，来说明 PLC 的硬件组成。

PLC 的基本结构如图 1-6 所示，由图 1-6 可知，由 PLC 作为控制器的自动控制系统，就是工业计算机控制系统。PLC 的中央处理器是由微处理器、单片机或位片式计算机组成的，且具有各种功能的 I/O 接口及存储器。下面结合图 1-6 说明 PLC 各个组成部分的功能。

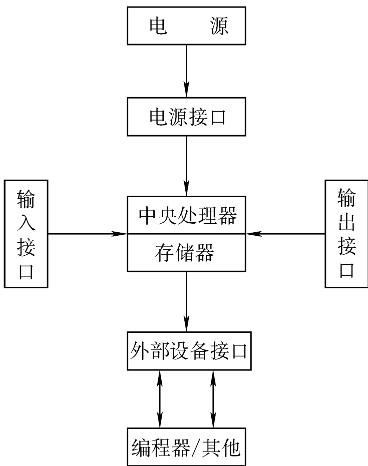


图 1-6　PLC 的基本结构

（1）中央处理器（CPU） 我们大家都知道，中央处理器（CPU）是计算机的核心，因此它也是 PLC 的核心。它按照系统程序（操作系统）赋予的功能完成的主要任务是：

1）接收与存储用户由编程器输入的用户程序和数据；

2）检查编程过程中的语法错误，诊断电源及 PLC 内部的工作故障；

3）用扫描方式工作，接收来自现场的输入信号，并输入到输入映像寄存器和数据存储器中；

4）在进入运行方式后，从存储器中逐条读取并执行用户程序，完成用户程序所规定的逻辑运算、算术运算及数据处理等操作；

5）根据运算结果更新有关标志位的状态，刷新输出映像寄存器的内容，再经输出部件实现输出控制、打印制表或数据通信等功能。

随着大规模集成电路的发展，PLC 采用单片机做 CPU 的越来越多，它以高集成度、高可靠性、高功能、高速度及低价格的优势，正在占领小型 PLC 的市场。

目前，小型 PLC 均为单 CPU 系统，而大、中型 PLC 通常是双 CPU 或多 CPU 系统。所谓双 CPU，是在 CPU 模板上装有两个 CPU 芯片，一个作为字处理器，一个作为位处理器。字处理器是主处理器，它执行所有的编程器接口的功能，监视内部定时器及扫描时间，完成字节指令的处理，并对系统总线和微处理器进行控制。位处理器是从处理器，它主要完成对位指令的处理，以减轻字处理器的负担。提高位指令的处理速度，并将面向控制过程的编程语言（如梯形图、流程图）转换成机器语言。

（2）存储器 PLC 的存储器中配有两种存储系统，即用于存放系统程序的系统程序存储器和存放用户程序的用户程序存储器。

系统程序存储器主要用来存储 PLC 内部的各种信息。一般系统程序是由 PLC 生产厂家编写的系统监控程序，不能由用户直接存取。系统监控程序主要由有关系统管理解释指令、标准程序及系统调用等程序组成。系统程序存储器一般用 PROM 或 EPROM 构成。

用户程序是由用户编写的程序，也称为应用程序。用户程序存放在用户程序存储器中，用户程序存储器的容量不大，主要存储 PLC 内部的输入输出信息，以及内部继电器、移位寄存器、累加寄存器、数据寄存器、定时器和计数器的动作状态。小型 PLC 的存储容量一般只有几千字节的容量（不超过 8KB）。我们一般讲 PLC 的内存大小，是指用户程序存储器的容量，用户程序存储器常用 RAM 构成。

用户程序存储器一般分为两个区，程序存储区和数据存储区。程序存储区用来存储由用户编写的、通过编程器输入的程序。而数据存储区用来存储通过输入端子读取的输入信号的状态、准备通过输出端子输出的输出信号的状态、PLC 中

各个内部器件的状态，以及特殊功能要求的有关数据。PLC 存储器的存储结构如表 1-1 所示。

表 1-1　PLC 存储器的存储结构

存储器	存储内容	
系统程序存储器	系统监控程序	
用户程序存储器	程序存储区	用户程序（如梯形图、语句表等）
	数据存储区	I/O 及内部器件的状态

（3）输入部件及接口（数字量）　来自现场的主令元件、检测元件的信号经输入接口进入到 PLC。主令元件的信号多数是指控制按钮，这种信号的特点基本上都是人为操作的。检测元件的信号主要来自各种传感器、限位开关、继电器等的触点。也可以说是过程控制当中某些位置变化或参数值变化所产生的信号。这些信号有的是开关量（数字量），有的是模拟量（连续变化的量），有的是直流信号，有的是交流信号，要根据输入信号的类型选择合适的输入接口。

为提高系统的抗干扰能力，各种输入接口均采取了抗干扰措施，如在输入接口内带有光耦合电路，使 PLC 与外部输入信号进行隔离。为消除信号噪声在输入接口内还设置了多种滤波电路。为便于 PLC 的信号处理，输入接口内有电平转换及信号锁存电路。为便于现场信号的连接，在输入接口的外部设有接线端子排。

（4）输出部件及接口（数字量）　由 PLC 产生的各种输出控制信号经输出接口去控制和驱动负载（如接触器和继电器线圈、电磁阀、指示灯、报警器等）。

因为 PLC 的直接输出带负载能力有限。最高电压为交流 220V，电流最高 2A，所以 PLC 输出接口所带的负载，通常就是上面这些。

同输入接口一样，输出接口的负载有交流的，也有直流的，要根据负载性质选择合适的输出接口。

输出接口的输出方式分为晶体管输出型，双向晶闸管输出型及继电器输出型。晶体管适用直流负载或 TTL 电路，双向晶闸管适用交流负载，而继电器输出型，既可用于直流负载，又可用于交流负载。使用时，只要外接一个与负载相符的电源即可。因而采用继电器输出型，对用户显得方便和灵活，但由于它是有触点输出，所以它的工作频率不能很高，工作寿命不如无触点的半导体器件长。

晶体管输出型每个输出点的最大带负载能力约为 0.75A，其接口响应速度较快，特别适合控制步进电动机之类的直流脉冲型负载。

双向晶闸管输出型每个点最大带负载能力约为 0.5～1A，其接口响应速度较快，适合控制要求频繁动作的交流负载。

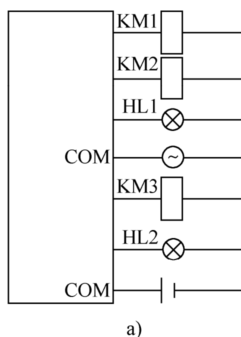
继电器输出型每个输出点的最大带负载能力约为 2A，作为数字量输出选择继电器型则更为自由和方便，且适用场合普遍。因此，在对动作时间和动作频率要

求不高的情况下，常常采用此方式。

输出接口的接线方式，PLC 的控制信号经输出接口送出来，接口与外部用户设备该如何接线呢？这里可分为两种方式，汇点式输出接线和隔离式输出接线。

① 汇点式输出接线方式，把所有的输出点分成几个组，一组一个公共端（COM），这样做的目的就是为了用户设备不同电压等级的要求，同级别电压的设备可放在同一组，通过本组的公共端（COM）形成电压回路如图 1-7a 所示，还可以将全部输出点作为一组，所有输出点共用一个 COM 点，如图 1-7b 所示。可以明显看出图 1-7b 的接线方式就不如图 1-7a 的接线方式灵活方便，图 1-7a 可以当作图 1-7b 使用，而图 1-7b 是不能当作图 1-7a 来使用的。

② 隔离式输出接线方式，隔离式输出接线方式如图 1-8 所示。在这种方式中，每个输出点都有自己的 COM 点，更加灵活方便，适合多级别电压的控制系统，每个 COM 点都是相对独立的，相互隔离的，如用户设备电压等级并不多，我们可给它们任意组合，做起来也很方便，只要把它们的 COM 点相应接在一起就可以组成同一电压等级回路。



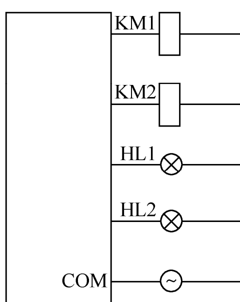


图 1-7　汇点式输出接线方式

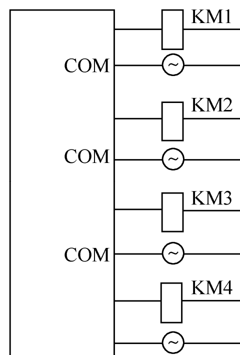


图 1-8　隔离式输出接线方式

（5）模拟量输入/输出接口模块　小型 PLC 的主机上一般没有模拟量输入/输出接口，如果有的话，通道数也是有限的，只有通过连接专用的扩展模块，才能增加模拟量通道数量。而大、中型 PLC 模拟量通道数可以达到成百上千个。

PLC 发展到今天，不仅仅只是处理数字量信号（开关量信号、断续量信号），也能够处理模拟量信号（连续变化的信号）。像温度、压力、速度、位移、电流、电压等信号，在生产设备上这些量都是连续变化的。以往这种信号都是通过仪表来控制，现在用 PLC 就完全可以实现控制。模拟量输入信号或模拟量输出信号可以是电压，也可以是电流，可以是单极性的，如 0～5V，0～10V，1～5V，4～20mA；也可以是双极性的，如±50mV，±5V，±10V，±20mA。

一个模拟 I/O 接口模块的通道数，可能有 2，4，8，16 个。也有的模块既有

输入通道，也有输出通道。

模拟量输入接口模块的任务是把现场中被测的模拟量信号转变成 PLC 可以处理的数字量信号。通常生产现场可能有多路模拟量信号需要采集，各模拟量的类型和参数都可能不同，这就需要在进入模块前，对模拟量信号进行转换和预处理，把它们转变成输入模块能统一处理的电信号，经多路转换开关进行多中选一，再将已选中的那路信号进行 A/D（模/数）转换，转换结束进行必要处理后，送入数据总线供 CPU 存取，或存入中间寄存器备用。

模拟量输出接口模块的任务是将 CPU 送来的数字量转换成模拟量，用以驱动执行机构，实现对生产过程或装置的闭环控制。

（6）扩展接口与通信接口　PLC 主机上除了输入接口与输出接口外，还有扩展接口与通信接口。

1）扩展接口现在有两个含义：一个是 CPU 的扩充，它是在原系统中只有一块 CPU 而无法满足系统工作要求时使用的，这个接口的功能是实现扩充 CPU，以及扩充 CPU 模块之间（多个 CPU 模块扩充）的相互控制和信息交换。另一个含义是单纯的 I/O（数字量 I/O 或模拟量 I/O）扩展接口，它是为弥补主机上 I/O 点数有限而设置的，用于扩展输入/输出点数，当用户的 PLC 控制系统所需的 I/O 点数超过主机本身的 I/O 点数时，就要通过 I/O 扩展接口将主机与 I/O 扩展单元连接起来，以满足用户的需求。

2）通信接口是专用于数据通信的，主要实现“人-机”对话或“机-人”对话。PLC 通过通信接口可与打印机监视器相连，也可与其他的 PLC 或上位计算机相连，构成多机局部网络系统或多级分布式控制系统，还有可实现管理与控制相结合的综合系统，用户应根据不同的设备要求遵循已规范好的通信协议，选择相应的通信方式并配置适合的通信接口，通信接口有串行接口和并行接口两种。

（7）编程器　编程器是人们以往最常用的编程设备，它又分为简易编程器和智能编程器，用它可以进行用户程序的输入、编辑、调试和监视，还可以通过其键盘去调用和显示 PLC 的一些内部继电器状态和系统参数，它使用 PLC 上专用接口与 CPU 联系，完成人机对话。通过简易编程器输入程序还必须把编好的梯形图程序转变成编程器认可的助记符，用于逐条将程序敲进去，需要很长的时间。编程器一般由 PLC 生产厂家提供，同一厂家的不同规格的 PLC 所使用的编程器是不一样的。

由 PLC 生产厂家生产的专用编程器使用范围有限，价格一般也较高，在个人计算机不断更新换代的今天，出现了使用以个人计算机为基础的编程系统。由生产厂家向用户提供编程软件，而编程软件装在哪台计算机则由用户自己选择。只要能支持此软件运行就可以了，用户在计算机上直接编写梯形图，然后下载到 PLC 中进行调试，还可以通过计算机监视程序运行。这种方法的主要优点就是利用个

人计算机，用几秒钟的时间就可以将程序输入到 PLC 当中去，能节省很多劳动和时间。监视运行方便、快捷、直观。对于不同厂家和型号的 PLC，只需要更换编程软件就可以了。

（8）其他　我们在进行 PLC 控制系统设计时还可以根据需要配置一些外部设备。

1）人-机接口装置（HMI）。人-机接口装置又叫操作员接口，用于实现操作人员与 PLC 控制系统的对话和相互作用。人-机接口最简单、最基本和最普遍的形式是由安装在控制台上的按钮、转换开关、拨码开关、指示灯、LED 数字显示器和声光报警等元件组成。它们用来指示 PLC 的 I/O 系统状态及各种信息。通过合理的程序设计，PLC 控制系统可以接收并执行操作员的指令，小型 PLC 一般采用这种人-机接口。

2）外存储器。PLC 的 CPU 内的半导体存储器称为内存，可用来存放系统程序和用户程序。有时将用户程序存储在盒式磁带机的磁带或磁盘驱动器的磁盘中，作为程序备份或改变生产工艺流程时调用。磁带和磁盘称为外存，如果 PLC 内存中的用户程序丢失或被破坏，可再次将存储在外存中的程序重新装入。

3）打印机。打印机在用户程序编制阶段用来打印带注解的梯形图程序或语句表程序，这些程序对用户的维修及系统的改造或扩展是非常有价值的。在系统的实时运行过程中，打印机用来提供运行过程中发生事件的硬记录，例如用于记录系统运行过程中报警的时间和类型。这对于分析事故原因和系统改进是非常重要的。在日常管理中，打印机可以定时或非定时打印各种生产报表。

二、PLC 的工作原理

众所周知，继电器控制系统是一种“硬件逻辑系统”，它所采用的是并行工作方式，也就是条件一旦形成，多条支路可以同时动作。PLC 是在继电器控制系统逻辑关系基础上发展演变的。而 PLC 是一种专用的工业控制计算机，其工作原理是建立在计算机工作原理基础上的。为了可靠地应用在工业环境下，便于现场电气技术人员的使用和维护，它有着大量的接口器件，特定的监控软件，专用的编程器件。这样一来，不但其外观不像计算机，它的操作使用方法、编程语言及工作过程与计算机控制系统也是有区别的。

实现它的工作原理是通过执行反映控制要求的用户程序，PLC 的 CPU 是以分时操作方式来处理各项任务的。计算机在每一瞬间只能做一件事，所以程序的执行是按程序顺序依次完成相应位置上的动作，所以它属于串行工作方式。

1. PLC 控制系统的等效工作电路

PLC 控制系统的等效工作电路可以分为 3 部分，即输入部分、内部控制电路和输出部分。输入部分就是采集输入信号，输出部分就是系统的执行部件。这两

部分与继电器控制电路相同，内部控制电路就是用户所编写的程序，可以实现控制逻辑，用软件编程代替继电器电路的功能。其等效工作电路如图 1-9 所示。图示的梯形图是为输出侧负载编写的对应程序，因 Q0.1、Q0.3、Q0.4 端子上没有接负载，所以也就不用给它们编写程序了。

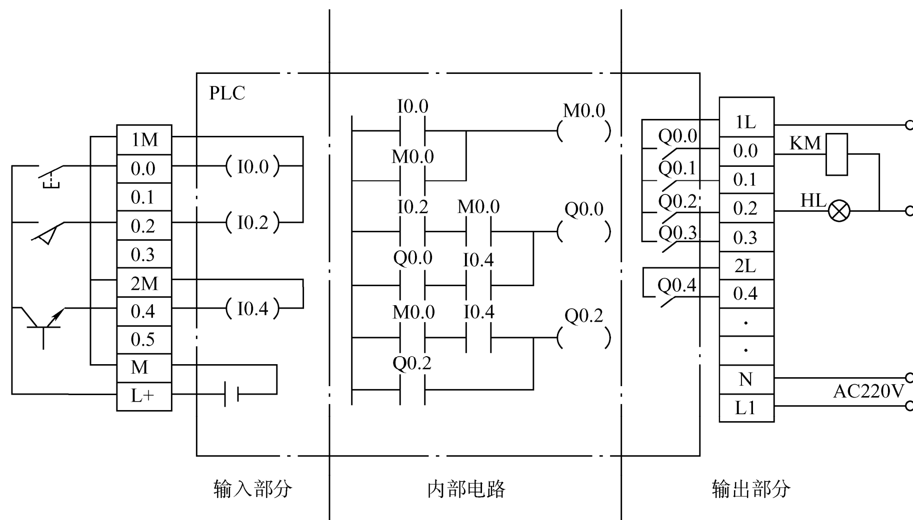


图 1-9　PLC 的等效工作电路

（1）输入部分　输入部分由外部输入电路、PLC 输入接线端子和输入继电器组成。外部输入信号经 PLC 输入接线端子去驱动输入继电器线圈。每个输入端子与其相同编号的输入继电器有着唯一确定的对应关系。当外部的输入元件处于接通状态时，对应的输入继电器线圈“得电”，这个输入继电器是 PLC 内部的软继电器，这样称呼也是便于用户接受，实际上这里不存在真正的物理上的继电器，只是存储器中的某一位，它可以提供任意多个动合触点或动断触点，这里所说的“触点”实际上也是不存在的，还是为了向早期的继电器线路图靠拢，便于用户接受。那么“触点”实际上就是这个存储器位的状态，这样一来就可以任意取用了。

为使输入继电器的线圈“得电”，即让外部输入元件的接通状态写入其对应的存储单元中去，输入回路要有电流，这个电源可以用 PLC 自己提供的 24V 直流电源，也可以由 PLC 外部的独立的交流或直流电源供电。

（2）内部控制电路　内部控制电路是由用户程序形成的用“软继电器”来替代硬继电器的控制逻辑。它的作用是按照用户编写的程序所规定的逻辑关系，处理输入信号和输出信号。一般用户程序是用梯形图语言编制的，它看上去很像继电器控制线路图，这也是 PLC 设计者所追求的。在前面我们已经提到过，即使 PLC

的梯形图与继电器控制线路图完全相同，最后的输出结果不一定相同，这是因为它们处理信号的过程是不一样的。继电器控制线路图中的继电器线圈都是并联关系，机会相等，只要条件允许可以同时动作，而 PLC 的梯形图程序的工作特点是周期性逐行扫描的，这样一来最后的输出结果就难免不一样了。

除了输入信号和输出信号，在 PLC 中还提供了计时器、计数器、辅助继电器（相当于继电器控制线路中的中间继电器）及某些特殊功能的继电器。为了实现我们的控制要求，在编程时可根据需要选用，但这些器件只能在 PLC 的内部控制电路中使用。在 PLC 的 I/O 点处是看不到它们的。

（3）输出部分（以数字量继电器输出型 PLC 为例）　输出部分是由在 PLC 内部且与内部控制电路隔离的输出继电器的动合触点、输出接线端子和外部驱动电路组成，用来驱动外部负载。

每个输出继电器除了有为内部控制程序提供编程用的任意多个动合、动断触点外，还为外部输出电路提供了一个实际的动合触点与输出接线端子相连。需要特别指出的是输出继电器是 PLC 中唯一存在的实际物理器件，打开 PLC 我们会发现在输出侧放置的那些微型继电器。

2. PLC 的工作原理

PLC 虽然具有许多微型计算机的特点，但它的工作原理却与微型计算机有很多不同点（这主要是各自的操作系统和系统软件的不同造成的）。

PLC 的工作原理有两个显著特点：一个是周期性顺序扫描，一个是信号集中批处理。

PLC 通电后，需要对软硬件都做一些初始化的工作，为了使 PLC 的输出及时地响应各种输入信号，初始化后反复不停地分步处理各种不同的任务，这种周而复始的循环工作方式称为周期性顺序扫描工作方式。

PLC 在运行过程中，总是处在不断循环的顺序扫描过程中，每次扫描所用的时间称为扫描时间，又称为扫描周期或工作周期。

由于 PLC 的 I/O 点数较多，采用集中批处理的方法可简化操作过程便于控制，提高系统可靠性。因此 PLC 的另一个主要特点就是对输入采样、执行用户程序、输出刷新实施集中批处理。

上面提到过 PLC 通电后，首先要进行的就是初始化工作，这一过程包括对工作内存的初始化，复位所有的定时器，将输入/输出继电器清零，检查 I/O 单元是否完好，如有异常则发出报警信号。初始化之后，就进入周期性扫描过程。小型 PLC 的工作过程如图 1-10 所示。

作为用户来讲，我们所关心的是怎样使用 PLC，PLC 怎样完成我们的控制要求，至于它内部的工作过程我们兴趣不大，所以在这里我们暂不分析初始化过程，那么扫描过程就只剩下“输入采样”，“执行用户程序”和“输出刷新”三个阶段

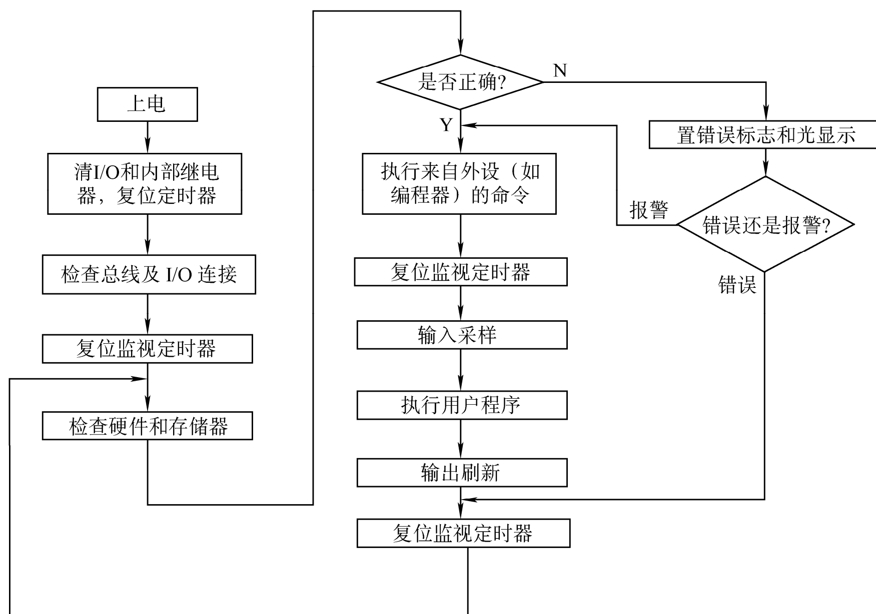


图 1-10 小型 PLC 的工作过程流程图

了。这三个阶段是 PLC 工作过程的中心内容，如图 1-11 所示。理解透 PLC 工作过程的这三个阶段是学习好 PLC 的基础，下面就详细分析这三个阶段：

（1）输入采样扫描阶段 在 PLC 的存储器中，设置了一片区域来存放输入信号和输出信号的状态，它们分别称为输入过程映像寄存器和输出过程映像寄存器，CPU 以字节（8 位）为单位来读写输入/输出过程映像寄存器。

这是第一个集中批处理过程，在这个阶段中，PLC 首先按顺序扫描所有输入端子，并将各输入状态存入相对应的输入映像寄存器中。此时，输入映像存储器被刷新，在当前的扫描周期内，用户程序依据的输入信号的状态（ON 或 OFF）均从输入映像寄存器中去读取，而不管此时外部输入信号的状态是否变化。在此程序执行阶段和接下来的输出刷新阶段，输入映像寄存器与外界隔离，即使此时外部输入信号的状态发生变化，也只能在下一个扫描周期的输入采样阶段去读取。一般来说，输入信号的宽度要大于一个扫描周期，否则很可能造成信号的丢失。

（2）执行用户程序的扫描阶段 PLC 的用户程序由若干条指令组成，指令在存储器中按照顺序排列。在 RUN 工作模式的程序执行阶段，在没有跳转指令时，CPU 从第一条指令开始，逐条顺序地执行用户程序。

在执行指令时，从 I/O 映像寄存器或别的位元件的映像寄存器读取其 ON/OFF 状态，并根据指令的要求执行相应的逻辑运算，运算的结果写入到相应的映像寄存器中。因此，除了输入映像寄存器属于只读的之外，各映像寄存器的内容随着

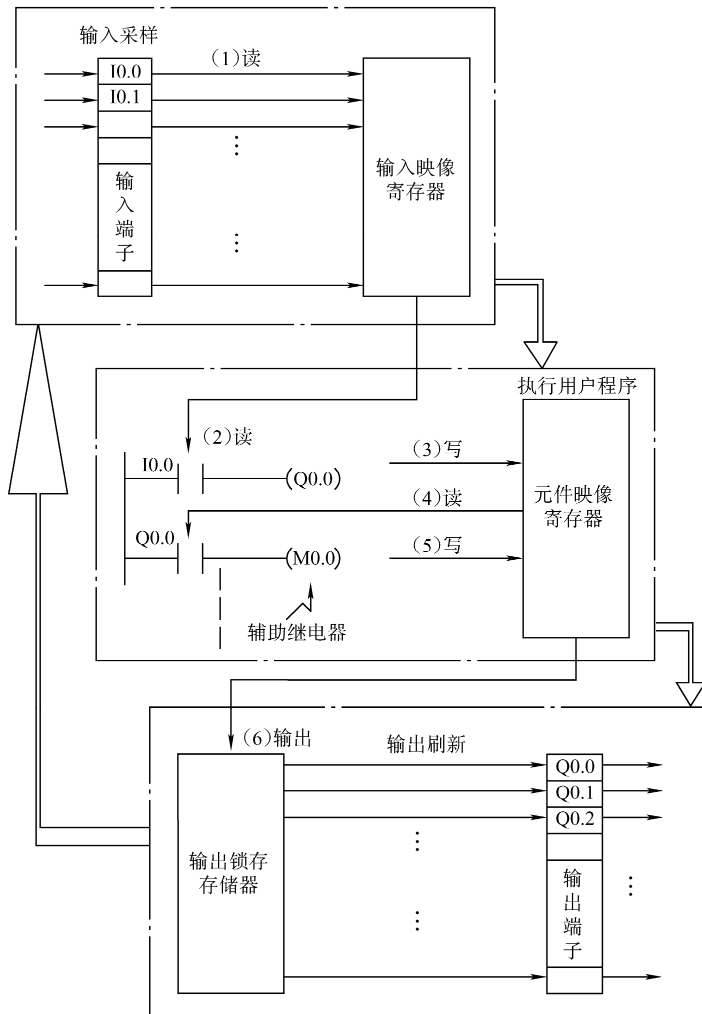


图 1-11　小型 PLC 的三个批处理过程

程序的执行而变化。

这是第二个集中批处理过程，具体地说在此阶段，PLC 的工作过程是这样的，CPU 对用户程序按顺序进行扫描，如果程序用梯形图表示，则总是按先上再下，从左至右的顺序进行扫描，每扫描到一条指令，所需要的输入信息的状态就要从输入映像寄存器中去读取，而不是直接使用现场的即时输入信息。因为第一个批处理过程（取输入信号状态）已经结束，“大门”已经关闭，现场即时信号此刻是进不来的。对于其他信息，则是从 PLC 的元件映像寄存器中读取，在这个过程顺序扫描中，每一次运算的中间结果都立即写入元件映像寄存器中，这样该元素的

状态马上就可以被后面将要扫描到的指令所利用，所以在编程时指令的先后位置将决定最后的输出结果。对输出继电器的扫描结果，也不是马上去驱动外部负载，而是将其结果写入元件映像寄存器中的输出映像寄存器中，同样该元素的状态也马上就可以被后面将要扫描到的指令所利用，待整个用户程序扫描阶段结束后，进入输出刷新扫描阶段时，成批地将输出信号状态送出去。

（3）输出刷新扫描阶段　CPU 执行完用户程序后，将输出映像寄存器的（ON/OFF）状态传送到输出模块并锁存起来，梯形图中某一输出位的线圈“得电”时，对应的输出映像寄存器为“1”状态。信号经输出模块隔离和功率放大后，继电器型输出模块中对应的硬件继电器（确实存在的物理器件）的线圈得电，它的常开触点闭合，使外部负载通电工作。到此，一个周期扫描过程中的三个主要过程就结束了，CPU 又进入了下一个扫描周期。

这是第三个集中批处理过程，用时极短，在本周期内，用户程序全部扫描后，就已经定好了某一输出位的状态，进入这段的第一步时，信号状态已送到输出映像寄存器中，也就是说输出映像寄存器的数据取决于输出指令的执行结果。然后再把此数据推到锁存器中锁存，最后一步就是锁存器的数据再送到输出端子上去。在一个周期中锁存器中的数据是不会变的。

PLC 的扫描工作过程如图 1-12 所示。

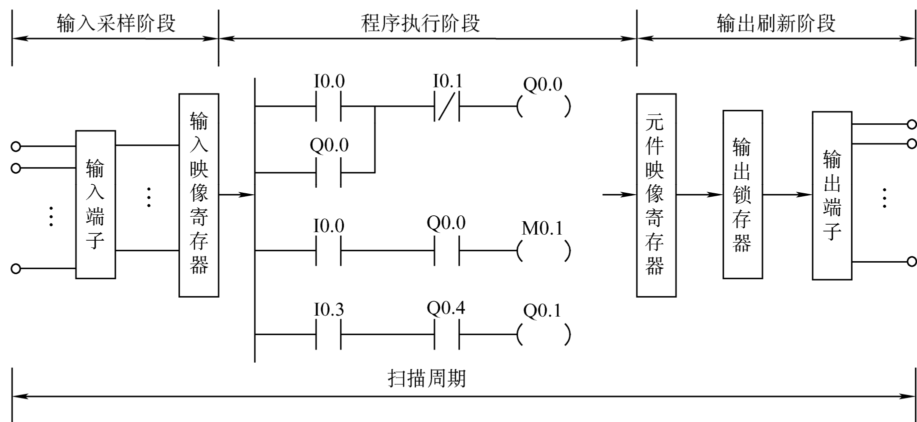


图 1-12　PLC 扫描工作过程

第五节　PLC 的应用设计及发展方向

一、PLC 的应用设计

PLC 控制系统是以程序形式来体现其控制功能的，大量的工作时间将用在软

件设计，也就是程序设计上。当我们接到控制要求后，首先要进行地址分配，画I/O分配图，但是这两项工作只需要很短的时间就可以完成，然后就要进行程序设计了。程序设计对于初学者来说通常采用继电器系统设计方法中的逐渐探索法，以步为核心，一步一步设计下去，一步一步修改调试，直到完成整个程序的设计。PLC 内部继电器很多，其触点在内存允许的情况下可重复使用，尽管初学者也许把程序设计得冗长，欠精练，但是由于整个程序扫描时间不过几十毫秒，只要能够准确地实现控制要求，达到控制目的，也就算完成了设计任务。归纳起来，PLC程序设计可遵循以下六步进行：

① 确定被控制系统必须完成的动作及完成这些动作的顺序；
② 分配输入/输出设备，即I/O分配，将信号与PLC的I/O口对应到位；
③ 设计PLC程序，通常是画梯形图；
④ 使用计算机编程软件，上机编写梯形图；
⑤ 将程序下载至PLC，进行调试（模拟或现场）；
⑥ 正常运行后保存已完成的程序。

显然，在建立一个PLC控制系统时，必须首先把系统需要的输入、输出数量确定下来，然后按照需要确定各种控制动作的顺序和各个控制装置彼此之间的相互关系。确定控制上的相互关系后，就可以进行编程的第二步——分配输入/输出设备，在分配了PLC的输入输出点、内部辅助继电器、定时器、计数器及特定功能块之后，就可以设计PLC程序画出梯形图。在画梯形图时要注意每个从左边母线开始的逻辑行必须终止于一个继电器线圈或定时器、计数器、功能块等，与实际的电路图是有区别的。梯形图画好后，使用编程软件直接把梯形图输入计算机并下载到PLC进行模拟调试，直至符合控制要求。这便是程序设计的整个过程。

二、PLC的发展方向

PLC 诞生不久即显示了其在工业控制中的重要地位，如日本、德国、法国等国家相继研制成功各自的PLC，受到工业界的欢迎。20世纪70年代末和80年代初，PLC 已成为工业控制领域中占主导地位的基础自动化设备。由最初的1位机发展为8位机，随着微处理器（CPU）和微型计算机技术在PLC中的应用，形成了现代意义上的PLC。现在的PLC产品已使用了16位、32位高性能微处理器，不仅控制功能增强，功耗、体积减小，可靠性提高，而且远程 I/O 和通信网络、数据处理以及人机界面（HMI）也有了长足的发展。如今，可编程序控制器技术已非常成熟。

目前，世界上有200多个厂家生产可编程序控制器产品，比较著名的厂家有美国的 AB、通用（GE）、莫迪康（MODICON），日本的三菱（MITSUBISHI）、欧姆龙（OMRON）、富士电机（FUJI）、松下电工，德国的西门子（SIEMENS），

法国的 TE、施耐德（SCHNEIDER），韩国的三星（SAMSUNG）、LG 等（其中 MODICON 和 TE 已归到 SCHNEIDER 旗下）。

PLC 总的发展趋势是向高集成度、小体积、大容量、高速度、易使用、高性能方向发展。具体表现在以下几个方面：

1. 向小型化、专业化、低成本方向发展

20 世纪 80 年代初，小型 PLC 在价格上还高于小系统用的继电器控制装置。随着微电子技术的发展，新型器件大幅度的提高功能和降低价格，使 PLC 结构更为紧凑，操作使用十分简便，功能不断增加。将原来大、中型 PLC 才有的功能部分地移植到小型 PLC 上，如模拟量处理、数据通信和复杂的功能指令等，但价格不断下降，真正成为现代电气控制系统中不可替代的控制装置。

2. 向大容量，高速度方向发展

随着自动化水平的不断提高，对中、大型机处理数据的速度要求也越来越高，在三菱公司的 32 位微处理器 M887788 中，在一块芯片上实现了 PLC 的全部功能，它将扫描时间缩短为每条基本指令 0.15μs。OMRON 公司的 CV 系列，每条基本指令的扫描时间为 0.125μs。SIEMENS 公司的 TI555 采用了多微处理器，每条基本指令的扫描时间为 0.068μs。大型 PLC 采用多微处理器系统，可同时进行多任务操作，处理速度提高，特别是增强了过程控制和数据处理的功能。另外，存储容量大大增加。

3. 智能型 I/O 模块的发展

智能型 I/O 模块是以微处理器和存储器为基础的功能部件，它们的 CPU 与 PLC 的主 CPU 并行工作，占用主 CPU 的时间很少，有利于提高 PLC 的扫描速度。它们本身就是一个小的微型计算机系统。智能 I/O 模块主要有模拟量 I/O、高速计数输入、中断输入、机械运动输入、热电偶输入、热电阻输入、条形码阅读器、多路 BCD 码输入/输出、模糊控制器、PID 回路控制和各种通信模块等。

4. 基于 PC 的编程软件取代编程器

随着计算机的日益普及，越来越多的用户使用基于个人计算机（PC）的编程软件。编程软件可以对 PLC 控制系统的硬件进行组态，即设置硬件的结构和参数，例如设置各框架各个插槽上模块的型号、模块的参数、各串行通信接口的参数等。在屏幕上可以直接生成和编辑梯形图、语句表、功能块图和顺序功能图程序，并可以实现不同编程语言的相互转换。程序被编译下载到 PLC，也可以将用户程序上传到计算机。程序可以存盘或打印，通过网络或 Modem 上网卡，还可以实现远程操作。

编程软件的调试和监控功能远远超过手持式编程器，例如在调试时可以设置执行用户程序的扫描次数，有的编程软件可以在调试程序时设置断点，有的具有跟踪功能，用户可以周期性地选择保存若干编程元件的历史数据，还可以将数据

上传后存为文件。

在 PC 上用编程软件编完程序后，要用专用电缆把 PC 与 PLC 连接起来，第一项需做的工作就是建立通信，也就是相互认识一下，在 PLC 运行过程中可以在梯形图中显示触点的通断和线圈的状态，查找复杂电路的故障非常方便。

5. PLC 编程语言的标准化与通信的易用化

与个人计算机相比，PLC 硬件、软件的体系结构都是封闭的而不是开放的。在硬件方面，各厂家的 CPU 模块和 I/O 模块互不通用。PLC 的编程语言和指令系统的功能和表达方式也不一致，因此各厂家的可编程序控制器互不兼容。为了解决这一问题，IEC（国际电工委员会）制定了可编程序控制器的编程语言标准。标准中除了提供几种编程语言供用户选择外，还允许编程者在同一程序中使用多种编程语言，这使得编程者能够选择不同的语言来适应特殊的工作。目前已有越来越多的工控产品厂商推出了符合标准的 PLC 指令系统或在 PC（个人计算机）上运行的软件包（软件 PLC）。

PLC 的通信联网功能使它能与个人计算机和其他智能控制设备交换数字信息，使系统形成一个统一的整体，实现分散控制和集中管理。通过双绞线、同轴电缆或光纤联网，信息可以传送到几十千米远的地方，通过 Modem 和互联网可以与世界上其他地方的计算机装置通信。

为了尽量减少用户在通信编程方面的负担，PLC 厂商做了大量的工作，使设备之间的通信自动地、周期性地进行，不需要用户为通信编程，用户的工作只是在组成系统时做一些硬件或软件上的初始化设置。

6. PLC 与现场总线相结合

IEC 对现场总线（Field Bus）的定义是：“安装在制造和过程区域的现场装置与控制室内的自动控制装置之间的数字式、串行、多点通信的数据总线称为现场总线”。它是当前工业自动化的热点之一。现场总线以开放的、独立的、全数字化的双向多变量通信代替 0～10mA 或 4～20mA 的现场电动仪表信号。现场总线 I/O 集检测、数据处理、通信于一体，可以代替变送器、调节器、记录仪等模拟仪表，它不需要框架、机柜，可以直接安装在现场导轨槽上。现场总线 I/O 的接线极为简单，只需一根电缆，从主机开始，沿数据链从一个现场总线 I/O 连接到下一个现场总线 I/O。使用现场总线后，自动控制系统的配线、安装、调试和维护等方面的费用可以节约 2/3 左右，现场总线 I/O 与 PLC 可以组成功能强大的、廉价的 DCS。

使用现场总线后，操作员可以在中央控制室实现远程监控，对现场设备进行参数调整，还可以通过现场设备的自诊断功能预测故障和寻找故障点。

第二章　S7-200 PLC 概述

S7-200 PLC 是德国西门子公司生产的一种小型 PLC，但其许多功能已经达到大、中型 PLC 的水平，而价格却与小型机一样，因此，它一经推出，即受到了广泛关注。特别是 S7-200 CPU22X 系列 PLC，由于它具有多种功能模块和人机界面（HMI）可供选择，所以系统的集成非常方便，还可以很容易地组成 PLC 网络，同时它具有功能齐全的编程和工业控制组态软件，使得在完成控制系统的设计时更加简单，几乎可以完成任何功能的控制任务。

德国的西门子（SIEMENS）公司是世界著名的、也是欧洲最大的电气设备制造商，是世界上研制、开发 PLC 较早的少数几个国家之一，欧洲第一台 PLC 就是西门子公司于 1973 年研制成功的，到 20 世纪末推出了 SIMATIC S7 系列 PLC，也可以说是它的第三代产品。

西门子公司的 PLC 在我国应用得十分普遍，尤其是大、中型 PLC，由于其可靠性高，在自动化控制领域中久负盛名。目前较先进的有 S7、M7 及 C7 三个系列的 PLC 产品，S7 系列的 PLC 根据控制系统规模的不同，分成三个系列：S7-200，S7-300，S7-400，分别对应小型、中型、大型 PLC。而基于 SIMATIC 系列 PLC 的各种功能模块、人机界面、工业网络、工业软件及控制方案的迅速发展，使 PLC 控制系统的功能更加强大，系统的设计和操作却越来越简便。

S7 系列 PLC 编程均使用 STEP 7 编程软件，该软件的 4.0 版已经完全汉化，编程非常方便，容易。考虑到西门子公司的产品在我国应用非常广泛，其功能比较全面和典型，具有一定的代表性，因此本章以 S7-200 CPU22X 系列为例，主要介绍以下内容：

① S7-200 的系统组成；
② S7-200 的性能特点及基本功能；
③ S7-200 的内部资源及寻址方式；
④ S7-200 的编程语言及程序结构。

第一节　S7-200 PLC 的系统组成

一、S7-200 的系统基本构成

S7-200 是西门子公司前几年投入市场的小型可编程序控制器，可以单机控制，

也可以进行输入/输出和功能模块的扩展。S7-200 属于整体式结构，它价格低廉，结构小巧，可靠性高，运行速度快，有极丰富的指令集，具有强大的多种集成功能和实时特性，具有很高的性能价格比。根据控制规模的大小（即输入/输出点数的多少），可以选择相应的 CPU 主机。除了 CPU221 以外，其他 CPU 主机均可进行系统扩展，在规模不太大的控制领域是较为理想的控制设备。

同其他的 PLC 一样，S7-200 的系统基本组成也是由主机单元加编程器组成。在需要进行系统扩展时，系统组成中还可包括：数字量扩展单元模块、模拟量扩展单元模块、通信模块、网络设备、人机界面（HMI）等。S7-200 的基本构成如图 2-1 所示。

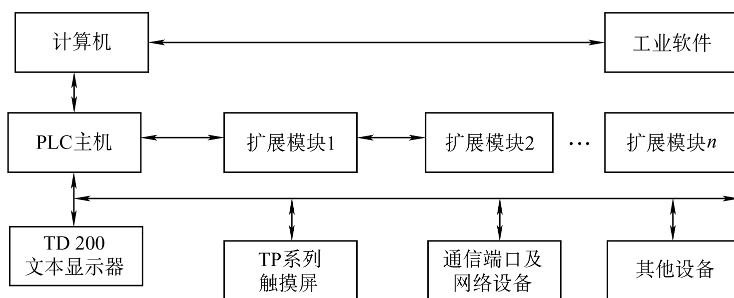


图 2-1　S7-200 PLC 系统的基本构成

二、主机单元

S7-200 的主机单元的 CPU 共有两个系列：CPU21X 及 CPU22X。CPU21X 系列包括 CPU212，CPU214，CPU215，CPU216；CPU22X 系列包括 CPU221，CPU222，CPU224，CPU226，CPU226XM。由于 CPU21X 系列属于 S7-200 的第一代产品，不再做具体介绍。

1. CPU221

① 6 输入/4 输出共 10 个数字量 I/O 点；

② 无 I/O 扩展能力；

③ 6KB 的程序和数据存储区空间；

④ 4 个独立的 30kHz 的高速计数器，2 路独立的 20kHz 的高速脉冲输出；

⑤ 1 个 RS-485 通信/编程口；

⑥ 具有多点接口（Multi Point Interface，MPI）通信协议；

⑦ 具有点对点接口（Point to Point Interface，PPI）通信协议；

⑧ 具有自由通信口。

2. CPU222

① 8 输入/6 输出共 14 个数字量 I/O 点；

② 可连接 2 个扩展模块单元，最大可扩展至 78 个数字量 I/O 点或 10 路模拟量 I/O；

③ 6KB 的程序和数据存储区空间；

④ 4 个独立的 30kHz 的高速计数器，2 路独立的 20kHz 的高速脉冲输出；

⑤ 具有 PID 控制器；

⑥ 1 个 RS-485 通信/编程口；

⑦ 具有多点接口（Multi Point Interface，MPI）通信协议；

⑧ 具有点对点接口（Point to Point Interface，PPI）通信协议；

⑨ 具有自由通信口。

3. CPU224

① 14 输入/10 输出共 24 个数字量 I/O 点；

② 可连接 7 个扩展模块单元，最大可扩展至 168 个数字量 I/O 点或 35 路模拟量 I/O；

③ 13KB 的程序和数据存储区空间；

④ 6 个独立的 30kHz 的高速计数器，2 路独立的 20kHz 的高速脉冲输出；

⑤ 具有 PID 控制器；

⑥ 1 个 RS-485 通信/编程口；

⑦ 具有多点接口（Multi Point Interface，MPI）通信协议；

⑧ 具有点对点接口（Point to Point Interface，PPI）通信协议；

⑨ 具有自由通信口；

⑩ I/O 端子排可以很容易地整体拆卸。

4. CPU226

① 24 输入/16 输出共 40 个数字量 I/O 点；

② 可连接 7 个扩展模板单元，最大可扩展至 256 个数字量 I/O 点或 35 路模拟量 I/O；

③ 13KB 的程序和数据存储区空间；

④ 6 个独立的 30kHz 的高速计数器，2 路独立的 20kHz 的高速脉冲输出；

⑤ 具有 PID 控制器；

⑥ 2 个 RS-485 通信/编程口；

⑦ 具有多点接口（Multi Point Interface，MPI）通信协议；

⑧ 具有点对点接口（Point to Point Interface，PPI）通信协议；

⑨ 具有自由通信口；

⑩ I/O 端子排可以很容易地整体拆卸。

5. CPU226MX

与 CPU226 相比，除了程序和数据存储区空间由 13KB 增加到 26KB 外，其余功能不变。

三、数字量扩展模块

S7-200 系列目前可以提供三大类共 9 种数字量输入/输出扩展模块。

（1）EM221，数字量输入（DI）扩展模块，具有 8 点 DC 输入，光耦合器隔离。

（2）EM222，数字量输出（D0）扩展模块，有 2 种输出类型：

① 8 点 DC24V 输出型；

② 8 点继电器输出型。

（3）EM223，数字量混合输入/输出（DI/DO）扩展模块，有 6 种输出类型：

① DC24V 输入 4 点/输出 4 点；

② DC24V 输入 4 点/继电器输出 4 点；

③ DC24V 输入 8 点/输出 8 点；

④ DC24V 输入 8 点/继电器输出 8 点；

⑤ DC24V 输入 16 点/输出 16 点；

⑥ DC24V 输入 16 点/继电器输出 16 点。

四、模拟量扩展模块

（1）EM231，4 路 12 位模拟量输入（AI）模块

① 差分输入，输入范围：电压：0～10V，0～5V，±2.5V，±5V；电流：0～20mA；

② 转换时间：<250μs；

③ 最大输入电压：DC30V，最大输入电流 32mA。

（2）EM232，2 路 12 位模拟量输出（AO）模块

① 输出范围：电压±10V，电流 0～20mA；

② 数据字格式：电压−32000～+32000，电流 0～+32000；

③ 分辨率：电压 12 位，电流 11 位。

（3）EM235，模拟量混合输入/输出（AI/AO）模块

① 模拟量输入 4 路，模拟量输出 1 路；

② 差分输入，电压：0～10V，0～5V，0～1V，0～500mV，0～100mV，0～50mV，±10V，±5V，±2.5V，±1V，±500mV，±250mV，±100mV，±50mV，±25mV；电流：0～20mA

③ 转换时间：<250μs；

④ 稳定时间：电压 100μs，电流 2ms。

五、智能模块

1. EM277 通信处理器

EM277 是连接 SIMATIC 现场总线 PROFIBUS-DP 从站的通信模块，使用 EM277 可以将 S7-200 PLC 的 CPU 作为现场总线 PROFIBUS-DP 的从站接到网络中。在 EM277 中，有一个 RS-485 接口，传输速率为 9.6kbit/s，19.2kbit/s，45.45kbit/s，93.75kbit/s，187.5kbit/s，500kbit/s～1Mbit/s，1.5Mbit/s，3Mbit/s，6Mbit/s，12Mbit/s，可自动设置。

2. CPU243-2 通信处理器

CPU243-2 是 S7-200 PLC（CPU 22X）的 AS-i 主站，通过连接 AS-i 可显著地增加 S7-200 PLC 的数字量输入/输出点数。每个主站最多可连接 31 个 AS-i 从站。S7-200 PLC 同时可以处理最多 2 个 CPU243-2，每个 CPU243-2 的 AS-i 上最大有 124DI/124DO。

六、其他设备

1. 编程设备（PG）

编程器是任何一台 PLC 不可缺少的设备，一般是由制造商专门提供的。S7-200 PLC 的编程器可以是简易的手持编程器 PG702，也可以是昂贵的图形编程器，如 PG740II，PG760II 等。为降低编程设备的成本，目前广泛采用个人计算机作为编程设备，但需配置制造商提供的专用编程软件。S7-200 PLC 的编程软件为 STEP-Micro/MIN32V4.0，通过一条 PC/PPI 电缆将用户程序送入 PLC 中。

2. 人机操作界面（Human Machine Interface，HMI）

（1）文本显示器 TD200　TD200 是 S7-200 PLC 的操作员界面，其功能如下：

① 显示文本信息。通过选择项确认的方法可显示最多 80 条信息，每条信息最多可包含 4 个变量。可显示中文；

② 设定实时时钟；

③ 提供强制 I/O 点诊断功能；

④ 可显示过程参数，并可通过输入键进行设定或修改；

⑤ 具有可编程的 8 个功能键，可以替代普通的控制按钮，从而可以节省 8 个输入点；

⑥ 具有密码保护功能。

TD200 不需要单独的电源，只需要将它的连接电缆接到 CPU22X 的 PPI 接口上，用 STEP7-Micro/WIN 软件进行编程。

（2）触摸屏 TP 070、TP 170A、TP 170B 及 TP 7、TP 27、TP 070、TP 170A、TP 170B 为具有较强功能且价格适中的触摸屏，其特点是：

① 在 Windows 环境下工作；
② 可通过 MPI 及 PROFIBUS-DP 与 S7-200 PLC 连接；
③ 背光管寿命达 50000h，可连续工作 6 年；
④ 利用 STEP7-Micro/WIN（Pro）SIMATIC ProTool/Lite V5.2 进行组态。
TP7 及 TP27 触摸屏主要是用于进行机床操作和监控。

第二节　S7-200 PLC 的性能特点及基本功能

一、S7-200 PLC 的主要技术性能指标

PLC 的技术性能指标是衡量其功能的直接反映，是设备选型的重要依据。S7-200 的 CPU22X 系列的主要技术性能指标见表 2-1。

表 2-1　CPU22X 系列的主要技术性能指标

指　　标	CPU221	CPU222	CPU224	CPU226
外形尺寸/mm	90×80×62	90×80×62	120.5×80×62	190×80×62
用户程序/字	2048	2048	4096	4096
用户数据/字	1024	1024	2560	2560
数据后备（电容）	50h	50h	50h	50h
本机 I/O	6 入/4 出	8 入/6 出	14 入/10 出	24 入/16 出
扩展模块数量	无	2 个	7 个	7 个
数字量 I/O 映像区	256	256	256	256
模拟量 I/O 映像区	无	16 入/16 出	32 入/32 出	32 入/32 出
布尔指令执行速度	0.37μs/指令	0.37μs/指令	0.37μs/指令	0.37μs/指令
FOR/NEXT 循环	有	有	有	有
整数指令	有	有	有	有
实数指令	有	有	有	有
I/O 映像寄存器	128I/128Q	128I/128Q	128I/128Q	128I/128Q
内部通用继电器	256	256	256	256
定时器/计数器	256/256	256/256	256/256	256/256
字入/字出	无	16/16	32/32	32/32
顺序控制继电器	256	256	256	256
内置高速计数器	4H/（20kHz）	4H/W（20kHz）	6H/W（20kHz）	6H/W（20kHz）
模拟电位器	1	1	2	2
直流脉冲输出	2（20kHz）	2（20kHz）	2（20kHz）	2（20kHz）
通信中断	1 发送/2 接收	1 发送/2 接收	1 发送/2 接收	2 发送/4 接收

（续）

指　　标	CPU221	CPU222	CPU224	CPU226
硬件输入中断	4 输入滤波器	4 输入滤波器	4 输入滤波器	4 输入滤波器
定时中断	2（1～255ms）	2（1～255ms）	2（1～255ms）	2（1～255ms）
实时时钟	有（时钟卡）	有（时钟卡）	有（内置）	有（内置）
口令保护	有	有	有	有
通信口数量	1（RS-485）	1（RS-485）	1（RS-485）	2（RS-485）
支持协议 0 号口 1 号口	PPI，DP/T 自由口 无	PPI，DP/T 自由口 无	PPI，DP/T 自由口 无	PPI，DP/T 自由口 同 0 号口
PROFIBUS 点对点	NETR/NETW	NETR/NETW	NETR/NETW	NETR/NETW

S7-200 PLC 的电源电压有 DC20.4～28.8V 和 AC85～264V 两种，主机上还集成了 24V 直流电源可以直接用于连接传感器等输入信号及小负荷执行机构。它的输出类型有专用于直流回路的晶体管型以及直流、交流都可以使用的继电器型两种输出方式。利用高速计数器专用指令及相对应的输入端可以捕捉比 CPU 扫描周期更快的脉冲信号，实现高速计数。两路最大可达 20kHz 的高频脉冲输出，可用以驱动步进电动机和伺服电动机以实现准确定位任务。可以用模块上的电位器来改变它对应的特殊寄存器中的数值，可以实时更改程序运行中的一些参数，如定时器/计数器的设定值，过程量的控制参数等。实时时钟可用于对信息加注时间标记记录机器运行时间或对过程进行时间控制。

二、S7-200 PLC 的输入/输出系统

PLC 通过输入/输出点与现场设备构成一个完整的 PLC 控制系统，因此要综合考虑现场设备的性质及 PLC 的输入/输出特性，才能更好地利用 PLC 的功能。

1. 输出特性

在 S7-200 PLC 中，输出信号有两种类型：继电器输出型和晶体管输出（DC 输出）型，CPU22X 的输出信号类型见表 2-2。

在表 2-2 中，电源电压是 PLC 的工作电压；输出电压是由用户提供的负载工作电压；每组点数是指全部输出端子可以分成几个隔离组，每个隔离组中有几个输出端子，例如：CPU226 中，4/5/7 表示共有 16 个输出端子分成 3 个隔离组，每个隔离组的输出端子数分别为 4 个，5 个，7 个，由于每个隔离组中有一个公共端，所以每个隔离组可以单独施加不同的负载工作电压。如果所有的负载工作电压相同，可将这些公共端连接起来。

2. 输入特性

在 S7-200 PLC 中，对数字量输入信号的电压要求均为 DC24V，“1”信号为

表 2-2 S7-200 PLC 的输出特性

CPU	类 型	电源电压	输出电压	输出点数	每组点数	输出电流
CPU221	晶体管	DC24V	DC24V	4	4	0.75A
	继电器	AC85～264V	DC24V，AC24～230V	4	1/3	2A
CPU222	晶体管	DC24V	DC24V	6	6	0.75A
	继电器	AC85～264V	DC24V，AC24～230V	6	3/3	2A
CPU224	晶体管	DC24V	DC24V	10	5/5	0.75A
	继电器	AC85～264V	DC24V，AC24～230V	10	4/3/3	2A
CPU226	晶体管	DC24V	DC24V	16	8/8	0.75A
	继电器	AC85～264V	DC24V，AC24～230V	16	4/5/7	2A

15～35V，“0”信号为 0～5V，经过光耦合器隔离后进入到 PLC 中。其特性见表 2-3。

表 2-3 S7-200 PLC 的输入特性

<table>
<tr><th>CPU</th><th>输入滤波</th><th>中断输入</th><th>高速计数器输入</th><th>每组点数</th><th>电缆长度</th></tr>
<tr><td>CPU221</td><td rowspan="4">0.2～12.8ms</td><td rowspan="4">I0.0～I0.3</td><td rowspan="4">I0.0～I0.5</td><td>2，4</td><td rowspan="4">非屏蔽输入 300m，屏蔽输入 500m，屏蔽中断输入及高速计数器 50m</td></tr>
<tr><td>CPU222</td><td>4，4</td></tr>
<tr><td>CPU224</td><td>8，6</td></tr>
<tr><td>CPU226</td><td>13，11</td></tr>
</table>

3. 输入/输出扩展能力

当主机单元模块上的 I/O 点数不够时，除了 CPU221 外，可以通过增加扩展单元模块的方法，对输入/输出点数进行扩展或增加模拟量控制。

在进行 I/O 扩展时，要考虑以下几个因素：

（1）CPU 主机模块所能连接的扩展模块数；

（2）CPU 主机模块的映像寄存器的数量；

（3）CPU 主机模块在 DC5V 下所能提供的最大扩展电流。

S7-200 PLC 的 CPU22X 系列的扩展能力见表 2-4。

表 2-4 S7-200 PLC 的扩展能力

CPU	最多扩展模块数	映像寄存器的数量	最大扩展电流/mA
CPU221	无	数字量：256，模拟量：无	0
CPU222	2	数字量：256，模拟量：16 入/16 出	340
CPU224	7	数字量：256，模拟量：32 入/32 出	660
CPU226	7	数字量：256，模拟量：32 入/32 出	100

S7-200 PLC 的 CPU22X 系列的扩展模块在 DC5V 下所消耗的电流见表 2-5。

表 2-5 S7-200 PLC 扩展模块的消耗电流

序 列	型 号	功 能	消耗电流/mA
1	EM221	数字量输入：8 点，晶体管输出	30
2	EM222	数字量输出：8 点，晶体管输出	50
3	EM222	数字量输出：8 点，继电器输出	40
4	EM223	数字量输入：4 点、输出：4 点，晶体管输出	40
5	EM223	数字量输入：4 点、输出：4 点，继电器输出	40
6	EM223	数字量输入：8 点、输出：8 点，晶体管输出	80
7	EM223	数字量输入：8 点、输出：8 点，继电器输出	80
8	EM223	数字量输入：16 点、输出：16 点，晶体管输出	160
9	EM223	数字量输入：16 点、输出：16 点，继电器输出	150
10	EM231	模拟量输入：4 路，12 位	20
11	EM231	模拟量输入：热电偶，4 路	60
12	EM231	模拟量输入：热电阻，4 路	60
13	EM232	模拟量输出：2 路，12 位	20
14	EM235	模拟量输入：4 路、输出 1 路，12 位	30
15	EM277	连接 PROFIBUS-DP	150

例如：CPU224 提供的扩展电流为 660mA，可以有以下几种扩展方案：

① 4 个 EM233，DI16/DO16 晶体管/继电器模块和 2 个 EM221 DI8 晶体管模块，消耗的电流为（4×150+2×30）mA=660mA；

② 4 个 EM233，DI16/DO16 晶体管/继电器输出模块，1 个 EM222 DO8 晶体管模块，消耗的电流为（4×150+1×40）mA=640mA；

③ 4 个 EM233，DI16/DO16 晶体管输出模块，消耗的电流为 4×160mA=640mA。

4. 快速响应功能

S7-200 PLC 的快速响应功能如下。

（1）脉冲捕捉功能　利用脉冲捕捉功能，使得 PLC 可以使用普通端子捕捉到小于一个 CPU 扫描周期的短脉冲信号。

（2）中断输入　利用中断输入功能，使得 PLC 可以极快的速度对信号的上升沿做出响应。

（3）高速计数器　S7-200 PLC 中有 4～6 个可编程的 30kHz 高速计数器，多个独立的输入端允许进行加减计数，可以连接相位差为 90°的 A/B 相增量的编码

器。

（4）高速脉冲输出　可利用 S7-200 PLC 的高速脉冲输出功能，驱动步进电动机或伺服电动机，实现准确定位。

（5）模拟电位器　模拟电位器的功能可用来改变某些特殊寄存器中的数值，这些特殊寄存器中的参数可以是定时器/计数器的设定值，或是某些过程变量的控制参数。在程序运行时利用模拟电位器，可随时更改这些参数，且不占用 PLC 的输入点。

5. 实时时钟

S7-200 PLC 的实时时钟用于记录机器的运行时间，或者对过程进行时间控制，以及对信息加注时间标记。

6. 功能扩展模块

当需要完成某些特殊功能的控制任务时，CPU 主机可以扩展特殊功能模块。如要求进行 PROFIBUS-DP 现场总线连接时，就需要 EM277PROFIBUS-DP 模块。

7. I/O 点数扩展和编址

CPU22X 系列的每种主机所提供的本机 I/O 点的 I/O 地址是固定的，进行扩展时，可以在 CPU 右边连接多个扩展模块，每个扩展模块的组态地址编号取决于各模块的类型和该模块在 I/O 链中所处的位置。编址方法是“按类分别排序”，也即同类型模块算是同组，在本组内编址排序，其他类型模块的有无以及所处的位置不影响本类型模块的编号。

例如，某一控制系统选用 CPU224，系统所需的输入输出点数各为：数字量输入 24 点、数字量输出 20 点、模拟量输入 6 点和模拟量输出 2 点。

本系统可有多种不同模块的选取组合，并且各模块在 I/O 链中的位置排列方式也可能有多种，如图 2-2 所示为其中的一种模块连接形式。表 2-6 所列为其对应的各模块的编址情况。

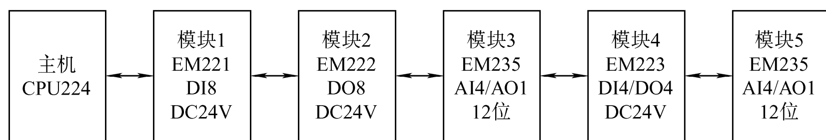


图 2-2　模块连接方式

由此可见，S7-200 PLC 系统扩展对输入/输出的组态规则为

① 同类型输入或输出点的模块进行顺序编址；

② 对于数字量，输入/输出映像寄存器的单位长度为 8 位（1 字节），本模块高位实际位数未满 8 位的，未用位不能分配给 I/O 链的后续模块；

③ 对于模拟量，输入/输出以 2 字节（1 个字）的递增方式来分配空间。

表 2-6 各模块编址

主机 I/O	模块 1 I/O	模块 2 I/O	模块 3 I/O	模块 4 I/O	模块 5 I/O
I0.0 Q0.0	I2.0	Q2.0	AIW0 AQW0	I3.0 Q3.0	AIW8 AQW2
I0.1 Q0.1	I2.1	Q2.1	AIW2	I3.1 Q3.1	AIW10
I0.2 Q0.2	I2.2	Q2.2	AIW4	I3.2 Q3.2	AIW12
I0.3 Q0.3	I2.3	Q2.3	AIW6	I3.3 Q3.3	AIW14
I0.4 Q0.4	I2.4	Q2.4			
I0.5 Q0.5	I2.5	Q2.5			
I0.6 Q0.6	I2.6	Q2.6			
I0.7 Q0.7	I2.7	Q2.7			
I1.0 Q1.0					
I1.1 Q1.1					
I1.2					
I1.3					
I1.4					
I1.5					

三、存储系统

S7 系列 PLC 中 CPU 的存储区组成如图 2-3 所示。

各个存储区的功能如下。

系统存储区
工作存储区
暂时局部存储区
程序存储区
累加器AC
地址寄存器

图 2-3 S7 系列 CPU 的存储区组成

1. 系统存储区

系统存储区（CPU 中的 RAM）用来存放操作数据，这些操作数据包括输入映像寄存器存储区的数据、输出映像寄存器存储区的数据、辅助继电器存储区的数据、定时器存储区的数据和计数器存储区的数据。

① 输入映像寄存器存储区用来存放输入状态值；

② 输出映像寄存器存储区用来存放经过程序处理的输出数据；

③ 辅助继电器存储区用来存放程序运行的中间结果；

④ 定时器存储区用来存放计时单元；

⑤ 计数器存储区用来存放计数单元。

2. 工作存储区

工作存储区（CPU 中的 RAM）用来存放 CPU 所执行的程序单元的复制件（逻辑块和数据块），还有为执行块调用指令而安排的暂时的局部变量存储区，该局部变量寄存器在块工作时一直保持，将块中的数据写入 L 堆栈中，数据只在块工作时有效，当调用新块时，L 堆栈重新分配。

3. 程序存储区

程序存储区可分成动态程序存储区（CPU 中的 RAM）和可选的固定程序存储区（EEPROM），用来存放用户程序。

4. 累加器

有 4 个 32 位的累加器（AC0～AC3），用来执行装载、传送、移位、算术运算等操作。

5. 地址寄存器

用来存放寄存器间接寻址的指针。

S7-200 PLC 的存储系统是由 RAM 和 EEPROM 组成的。在 CPU 模块内，配置了一定容量的 RAM 和 EEPROM，S7-200 PLC 的 CPU22X 的存储容量见表 2-7。

表 2-7　S7-200 PLC 的存储容量

CPU	用户程序存储区容量/字	用户数据存储区容量/字	用户存储器类型
CPU221	2048	1024	EEPROM
CPU222	2048	1024	EEPROM
CPU224	4096	2560	EEPROM
CPU226	4096	2560	EEPROM

当 CPU 主机单元模块的存储器容量不够时，可通过增加 EEPROM 存储器卡的方法扩展系统的存储容量。S7-200 PLC 的存储系统如图 2-4 所示。

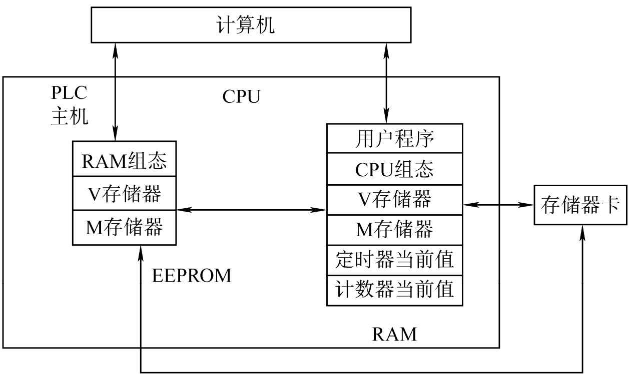


图 2-4　S7-200 PLC 的存储系统

S7-200 PLC 的程序结构一般由三部分组成：用户程序、数据块和参数块。用户程序是必不可少的，是程序的主体；数据块是用户程序在执行过程中所用到的和生成的数据；参数块是指 CPU 的组态数据。数据块和参数块是程序的可选部分。

存储系统的使用，主要有以下几个方面。

（1）设置保持数据的存储区　为了防止系统运行时突然掉电而导致一些重要

数据的丢失，可以在设置 CPU 组态参数时定义要保持数据的存储区。这些存储区包括变量存储器、通用辅助继电器、计数器和 TONR 型定时器。

（2）永久保存数据　通过对 S7-200 PLC 中的特殊标志存储器字节 SMB31 和存储器字 SMW32 的设置，可以实现将存储在 RAM 中变量存储器区任意位置的字节、字、双字数据备份到 EEPROM 存储器。

（3）存储器卡的使用　存储器卡的作用类似于计算机的软磁盘，可以将 PLC 中的 CPU 的组态参数、用户程序和存储在 EEPROM 中的变量存储器永久区的数据进行备份。

四、S7-200 PLC 的工作方式及扫描周期

1. 工作方式

S7-200 PLC 有 3 种工作方式：RUN（运行）、STOP（停止）、TERM（Terminal 终端）工作方式，可通过安装在 PLC 上的方式选择开关进行切换。

① STOP 方式：在 STOP 方式下，不能运行用户程序，可以向 CPU 装载用户程序或进行 CPU 的设置；

② TERM 方式：在 TERM 方式下，允许使用工业编程软件 STEP7-Micro/WIN32 来控制 CPU 的工作方式；

③ RUN 方式：在 RUN 方式下，CPU 执行用户程序。

当电源掉电又恢复后，如果方式选择开关在 TERM 或 STOP 状态下，CPU 自动进入 STOP 方式。如果方式选择开关在 RUN 状态下，则 CPU 自动进入 RUN 方式。

2. 扫描周期

在 RUN 方式下，系统周期性地循环执行用户程序。在每个扫描周期内，主要完成的任务如图 2-5 所示。

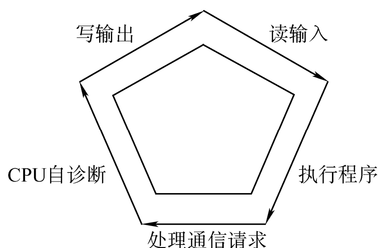


图 2-5　S7-200 PLC 的扫描周期

（1）读输入阶段（输入采样阶段）　在输入采样阶段，根据输入量的不同，所做的工作也不同。如果输入量是数字量，则在每个扫描周期的开始，先进行输入采样，将数字量输入点的当前值，写到输入映像寄存器中。如果输入量是模拟量，对于输入信号变化较慢的模拟量，则采用数字滤波，CPU 从模拟量输入模块读取滤波值；对于高速信号，一般不用数字滤波，CPU 直接读取模拟值。

对于需要利用模拟量控制字传递报警信息的模块，则不能使用模拟量的数字滤波功能，对于 RTD、热电偶及 AS-i 主站模块，禁止进行数字滤波。

（2）执行程序阶段　在执行程序阶段，对于普通的数字量输入和输出，CPU 以循环扫描的工作方式，从用户程序的第一条指令开始，执行到结束指令，完成

一个扫描周期，又进入下一个扫描周期，与图 2-5 所描述的扫描过程是一致的。而对于立即 I/O 指令、模拟量 I/O 指令和中断指令，则与图 2-5 所描述的扫描过程有所不同。

① 立即 I/O 指令：这是在程序中安排的对输入点的信息立即读取，或对输出点的状态立即刷新的指令，执行该指令时，不受扫描周期的约束；

② 模拟量 I/O 指令：对于不设数字滤波的直接模拟量的输入/输出，其执行方式与立即 I/O 指令基本相同；

③ 中断指令：如果在程序中使用了中断指令，则在处理中断事件时，中断子程序与主程序一起被存入存储器，进入 CPU 的扫描周期。中断程序的执行，增加了 CPU 的扫描周期，且使扫描周期变得不固定。在编制用户程序时，必须考虑到这一点。

（3）处理通信请求阶段　在处理通信请求阶段，CPU 自动检测来自各个通信端口的通信信息，并对这些信息进行自动处理。

（4）CPU 自诊断阶段　在 CPU 自诊断阶段，CPU 检测主机硬件，同时检查所有的 I/O 模块的状态。

（5）写输出阶段（输出刷新阶段）　在输出刷新阶段，CPU 用输出映像寄存器中的数据对输出点进行刷新。

第三节　S7-200 PLC 的内部资源及寻址方式

一、S7-200 PLC 的基本数据类型

在 S7-200 PLC 的编程语言中，大多数指令要同具有一定大小的数据对象一起进行操作。不同的数据对象具有不同的数据类型，不同的数据类型具有不同的数制和格式选择。程序中所用的数据可指定一种数据类型。在指定数据类型时，要确定数据大小和数据位结构。S7-200 PLC 的基本数据类型及范围见表 2-8。

表 2-8　S7-200 PLC 的基本数据类型及范围

基本数据类型	位数	说明
布尔型 BOOL	1	位　范围：0，1
字节型 BYTE	8	字节　范围：0～255
字型 WORD	16	字　范围：0～65535
双字型 DWORD	32	双字　范围：0～（2^{31}–1）
整型 INT	16	整数　范围：–32768～+32767
双整型 DINT	32	双字整数　范围：-2^{31}～（2^{31}–1）
实数型 REAL	32	IEEE 浮点数

在编程中经常会使用常数。常数数据长度可分为字节、字和双字。在机器内部的数据都以二进制存储，但常数的书写可以用二进制、十进制、十六进制、ASCII 或浮点数（实数）等多种形式。几种常数表示方法见表 2-9。

表 2-9 常数表示方法

进　制	书写格式	举　　例
十进制	十进制数值	1052
十六进制	16#十六进制值	16#8AC6
二进制	2#二进制值	2#1010_0011_1101_0001
ASCII	‘ASCII 码文本’	‘Show terminals’
浮点数	ANSI/IEEE 754—1985 标准	（正数）+1.175495E−38～+3.402823E+38
		（负数）−1.175495E−38～−3.402823E+38

注：表中的#为常数的进制格式说明符，如果常数无任何格式说明符，则系统默认为十进制数。

二、软元件（编程元件）

1. 软元件（软继电器）

用户使用的 PLC 中的每一个输入/输出、内部存储单元、定时器和计数器都称为软元件。各元件有其不同的功能，有固定的地址。软元件的数量决定了可编程序控制器的规模和数据处理能力，每一种 PLC 的软元件是有限的。

软元件是 PLC 内部具有一定功能的器件，这些器件实际上是由电子电路和寄存器及存储器单元等组成。例如，输入继电器是由输入电路和输入映像寄存器构成；输出继电器是由输出电路和输出映像寄存器构成；定时器和计数器也都是由特定功能的寄存器构成。它们都具有继电器特性，但没有机械性的触点。为了把这种元器件与传统电气控制电路中的继电器区别开来，我们把它们称为软元件或软继电器。这些软继电器的最大特点是其触点（包括常开触点和常闭触点）可以无限次使用，并且它们的寿命长。

编程时，用户只需要记住软元件的地址即可。每一个软元件都有一个地址与之相对应，软元件的地址编排采用区域号加区域内编号的方式，即 PLC 内部根据软元件的功能不同，分成了许多区域，如输入/输出继电器区、定时器区、计数器区、特殊继电器区等，分别用 I、Q、T、C、SM 等来表示。

可编程序控制器在其系统软件的管理下，将用户程序存储器（即装载存储区）划分出若干个区，并将这些区赋予不同的功能，由此组成了各种内部器件，这些内部器件就是 PLC 的编程元件。PLC 的编程元件的种类和数量是因不同厂家、不同系列、不同规格而异，编程元件的种类及数量越多，其功能就越强。这些编程元件沿用了传统继电器控制电路中继电器的名称，并根据其功能，分别称为输入

继电器、输出继电器、辅助继电器、变量继电器、定时器、计数器、数据寄存器等。

需要说明的是，在 PLC 内部，并不真正存在这些物理器件，与其对应的只是存储器中的某些存储单元。一个继电器对应一个基本单元（即 1 位，1bit），多个继电器将占有多个基本单元；8 个基本单元形成一个 8 位二进制数，通常称为 1 字节（1Byte），它正好占用普通存储器的一个存储单元，连续两个存储单元构成一个 16 位二进制数，通常又称为一个字（Word），或一个通道。连续的两个通道还能构成所谓的双字（Double Words）。各种编程元件，各自占有一定数量的存储单元。使用这些编程元件，实质上就是对相应的存储内容以位、字节、字（或通道）或双字的形式进行存取。

2. 软元件介绍

（1）输入继电器（I）　输入继电器就是 PLC 的存储系统中的输入映像寄存器。它的作用是接收来自现场的控制按钮、行程开关及各种传感器等的输入信号。通过输入继电器，将 PLC 的存储系统与外部输入端子（输入点）建立起明确对应的连接关系，它的每 1 位对应 1 个数字量输入点。输入继电器的状态是在每个扫描周期的输入采样阶段接收到的由现场送来的输入信号的状态（“1”或“0”）。由于 S7-200 PLC 的输入映像寄存器是以字节为单位的寄存器，CPU 一般按“字节.位”的编址方式来读取一个继电器的状态，也可以按字节（8 位）或者按字（2B、16 位）来读取相邻一组继电器的状态。前面在介绍 PLC 的等效工作电路时已强调过，不能通过编程的方式改变输入继电器的状态，但可以在编程时，通过使用输入继电器的触点，无限制地使用输入继电器的状态。如我们没在输入端子上接器件，那这个输入继电器只能空着，不能挪做它用。

（2）输出继电器（Q）　输出继电器就是 PLC 存储系统中的输出映像寄存器。通过输出继电器，将 PLC 的存储系统与外部输出端子（输出点）建立起明确对应的连接关系。S7-200 PLC 的输出继电器也是以字节为单位的寄存器，它的每 1 位对应 1 个数字量输出点，一般采用“字节.位”的编址方法。输出继电器的状态可以由输入继电器的触点、其他内部器件的触点，以及它自己的触点来驱动，即它完全是由编程的方式决定其状态。我们也可以像使用输入继电器触点那样，通过使用输出继电器的触点，无限制地使用输出继电器的状态。输出继电器与其他内部器件的一个显著不同在于它有一个且仅有一个实实在在的物理动合触点，用来接通负载。这个动合触点可以是有触点的（继电器输出型），或者是无触点的（晶体管输出型或双向晶闸管输出型）。没有使用的输出继电器，可以做内部继电器使用，但一般不推荐这种用法，这种用法可能引起不必要的误解。

输出继电器 Q 的线圈一般不能直接与梯形图的逻辑母线连接，如果某个线圈确实不需要经过任何编程元件触点的控制，可借助于特殊继电器 SM0.0 的动合触

点。

（3）变量寄存器（V） S7-200 PLC 中有大量的变量寄存器，用于模拟量控制、数据运算、参数设置及存放程序执行过程中控制逻辑操作的中间结果。变量寄存器可以位为单位使用，也可按字节、字、双字为单位使用。变量寄存器的数量与 CPU 的型号有关，CPU222 为 V0.0～V2047.7，CPU224 为 V0.0～V5119.7，CPU226 为 V0.0～V5119.7。

（4）辅助继电器（M） 在逻辑运算中，经常需要一些辅助继电器，它的功能与传统的继电器控制电路中的中间继电器相同。辅助继电器与外部没有任何联系，不可能直接驱动任何负载。每个辅助继电器对应着数据存储区的一个基本单元，它可以由所有的编程元件的触点（当然包括它自己的触点）来驱动。它的状态同样可以无限制使用。借助于辅助继电器的编程，可使输入输出之间建立复杂的逻辑关系和联锁关系，以满足不同的控制要求。在 S7-200 PLC 中，有时也称辅助继电器为位存储区的内部标志位（Marker），所以辅助继电器一般以位为单位使用，采用“字节.位”的编址方式，每 1 位相当 1 个中间继电器，S7-200 PLC 的 CPU22X 系列的辅助继电器的数量为 256 个（32B，256 位）。辅助继电器也可以字节、字、双字为单位，作存储数据用。建议用户存储数据时使用变量寄存器（V）。

（5）特殊继电器（SM） 特殊继电器用来存储系统的状态变量及有关的控制参数和信息。它是用户程序与系统程序之间的界面，用户可以通过特殊寄存器来沟通 PLC 与被控对象之间的信息，PLC 通过特殊继电器为用户提供一些特殊的控制功能和系统信息，用户也可以将对操作的特殊要求通过特殊继电器通知 PLC。例如可以读取程序运行过程中的设备状态和运算结果信息，利用这些信息实现一定的控制动作。用户也可以通过对某些特殊继电器的直接设置，使设备实现某种功能。

CPU22X 系列 PLC 的特殊继电器是 SM0.0～SM299.7。

对 SMB0：有 8 个状态位。在每个扫描周期的末尾，由 S7-200 PLC 的 CPU 更新这 8 个状态位。因此这 8 个 SM 为只读型 SM，这些特殊继电器的功能和状态是由系统软件决定的，与输入继电器一样，不能通过编程的方式改变其状态，只能利用这些特殊继电器的功能和状态。

（6）计数器（C） 计数器也是广泛应用的重要编程元件，用来对输入脉冲的个数进行累计，实现计数操作。使用计数器时要事先在程序中给出计数的设定值（也称预置值，即要进行计数的脉冲数）。当满足计数器的触发输入条件时，计数器开始累计计数输入端的脉冲前沿的次数，当达到设定值时，计数器动作。CPU22X 系列的 PLC 共有 256 个计数器，其编号为 C0～C255。每个计数器都有一个 16 位的当前寄存器及 1 个状态位 C-bit。

计数器包含两方面的信息，计数器当前值和计数器状态位。

计数器状态位：当计数器的当前值达到设定值时，C-bit 为“ON”。

计数器当前值：在计数器当前值寄存器中存储的当前所累计的脉冲个数，用 16 位符号整数表示。

计数器指令中所存取的是计数器当前值还是计数器状态位，取决于所用的指令，带位操作的指令存取计数器状态位，带字操作的指令存取计数器的当前值。

计数器的计数方式有 3 种，递增计数、递减计数和增/减计数。递增计数是从 0 开始，累加到设定值，计数器动作。递减计数是从设定值开始，累减到 0，计数器动作。

PLC 的计数器的设定值和定时器的设定值，一般不仅可以用程序设定，也可以通过 PLC 内部的模拟电位器或 PLC 外界的拨码开关，方便、直观地随时修改。

（7）定时器（T）　定时器是 PLC 的重要编程元件，它的作用与继电器控制电路中的时间继电器基本相似。定时器的设定值通过程序预先输入，当满足定时器的工作条件时，定时器开始计时，定时器的当前值从 0 开始按照一定的时间单位（即定时精度）增加，例如对于 10ms 定时器，定时器的当前值间隔 10ms 加 1。当定时器的当前值达到它的设定值时，定时器动作。

CPU22X 系列 PLC 的定时器数量为 256 个，T0～T255。定时器的定时精度分别为 1ms，10ms 和 100ms，1ms 的定时器有 4 个，10ms 的定时器有 16 个，100ms 的定时器有 236 个。这些定时器的类型可分为 3 种：接通延时型定时器 TON、断开延时型定时器 TOF、保持型接通延时定时器 TONR。CPU22X 系列 PLC 定时器的定时精度及编号见表 2-10。

表 2-10　CPU22X 定时器的精度及编号

定时器类型	定时精度/ms	最大当前值/s	定时器号
TON TOF	1	32.767	T32，T96
	10	327.67	T33～T36，T97～T100
	100	3276.7	T37～T63，T101～T255
TONR	1	32.767	T0，T64
	10	327.67	T1～T4，T65～T68
	100	3276.7	T5～T31，T69～T95

在使用定时器时要注意，不能把一个定时器号同时用做 TON 和 TOF，例如：在一个程序中既有 TON T32，又有 TOF T32。

定时器号包含两方面的信息，定时器当前值和定时器状态位，每个定时器都有一个 16 位的当前值寄存器，以及 1 个状态位 T-bit，如图 2-6 所示。

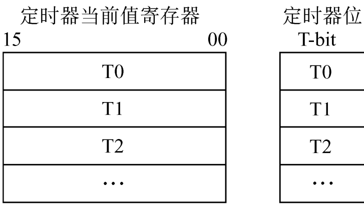


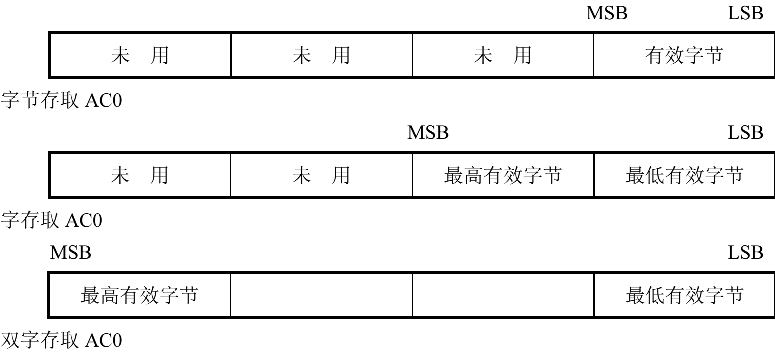
图 2-6 PLC 中的定时器

定时器状态位：当定时器的当前值达到设定值时，T-bit 为“ON”。

定时器当前值：在定时器当前值寄存器中存储的当前所累计的时间，用 16 位符号整数表示。定时器指令中所存取的是定时器当前值还是定时器状态位，取决于所用的指令，带位操作的指令存取定时器状态位，带字操作的指令存取定时器的当前值。

（8）高速计数器（HSC） 普通计数器的计数频率受扫描周期的制约，在需要高频计数的情况下，可使用高速计数器。与高速计数器对应的数据，只有一个高速计数器的当前值，是一个带符号的 32 位的双字型数据。

（9）累加器（AC） 累加器是可像存储器那样使用的读/写设备，是用来暂存数据的寄存器，它可以向子程序传递参数，或从子程序返回参数，也可以用来存放运算数据、中间数据及结果数据。S7-200 PLC 共有 4 个 32 位的累加器：AC0～AC3。使用时只表示出累加器的地址编号（如 AC0）。累加器存取数据的长度取决于所用的指令，它支持字节、字、双字的存取，以字节或字为单位存取累加器时，是访问累加器的低 8 位和低 16 位。



（10）状态继电器（S，也称为顺序控制继电器） 状态继电器是使用步进控制指令编程时的重要编程元件，用状态继电器和相应的步进控制指令，可以在小型 PLC 上编制较复杂的控制程序。

（11）局部变量存储器（L） 局部变量存储器用于存储局部变量。S7-200 PLC 中有 64 个局部变量存储器，其中 60 个可以用作暂时存储器或者给子程序传递参

数。如果用梯形图或功能块图编程，STEP 7-Micro/WIN32 保留这些局部变量存储器的最后 4B。如果用语言表编程，可以寻址到全部 64B，但不要使用最后 4B。

局部变量存储器与存储全局变量的变量寄存器很相似，主要区别是变量寄存器是全局有效的，而局部变量存储器是局部有效的。全局是指同一个存储器可以被任何一个程序（主程序、子程序、中断程序）读取，局部是指存储器区和特定的程序相关联。S7-200 PLC 给主程序分配 64 个局部变量存储器。给每级嵌套子程序分配 64B 局部变量存储器，给中断程序分配 64 个局部变量存储器。子程序不能访问分配给主程序、中断程序和其他子程序的局部变量存储器，子程序和中断程序不能访问主程序的局部变量存储器，中断程序也不能访问主程序和子程序的局部变量存储器。

S7-200 PLC 根据需要自动分配局部变量存储器。当执行主程序时，不给子程序和中断程序分配局部变量存储器，当出现中断或调用子程序时，才给子程序和中断程序分配局部变量存储器。新的局部变量存储器在分配时可以重新使用，分配给不同子程序或中断程序相同编号的局部变量存储器。

可以按位、字节、字、双字访问局部变量存储器，可以把局部变量存储器作为间接寻址的指针，但是不能作为间接寻址的存储器区。

（12）模拟量输入（AIW）寄存器/模拟量输出（AQW）寄存器　PLC 处理模拟量的过程是，模拟量信号经 A/D 转换后变成数字量存储在模拟量输入寄存器中，通过 PLC 处理后将要转换成模拟量的数字量写入模拟量输出寄存器，再经 D/A 转换成模拟量输出。即 PLC 对这两种寄存器的处理方式不同，对模拟量输入寄存器只能作读取操作，而对模拟量输出寄存器只能作写入操作。

由于 PLC 处理的是数字量，其数据长度是 16 位，因此要以偶数号字节进行编址，从而存取这些数据。

三、CPU 存储区域（软元件）的直接寻址

1. 直接寻址

S7-200 PLC 的存储单元按字节进行编址，无论所寻址的是何种数据类型，通常应指出它所在存储区域内的字节地址。每个单元都有唯一的地址，这种直接指出元件名称的寻址方式称为直接寻址。S7-200 PLC 软元件的直接寻址的符号见表 2-11。

表 2-11 中：

A：元件名称，即该数据在数据存储器中的区域地址，可以是表 2-11 中的元件符号；

T：数据类型，若为位寻址，则无该项；若为字节、字或双字寻址，则 T 的取值应分别为 B、W 和 D；

表 2-11　S7-200 PLC 元件名称及直接编址格式

元件符号（名称）	所在数据区域	位寻址格式	其他寻址格式
I（输入继电器）	数字量输入映像区	Ax.y	ATx
Q（输出继电器）	数字量输出映像区	Ax.y	ATx
M（通用辅助继电器）	内部存储器区	Ax.y	ATx
SM（特殊继电器）	特殊存储器区	Ax.y	ATx
S（顺序控制继电器）	顺序控制继电器存储器区	Ax.y	ATx
V（变量存储器）	变量存储器区	Ax.y	ATx
L（局部变量存储器）	局部存储器区	Ax.y	ATx
T（定时器）	定时器存储器区	Ax	Ax（仅字）
C（计数器）	计数器存储器区	Ax	Ax（仅字）
AI（模拟量输入映像寄存器）	模拟量输入存储器区	无	Ax（仅字）
AQ（模拟量输出映像寄存器）	模拟量输出存储器区	无	Ax（仅字）
AC（累加器）	累加器区	无	Ax（任意）
HSC（高速计数器）	高速计数器区	无	Ax（仅双字）

x：字节地址；

y：字节内的位地址，只有位寻址才有该项。

2. 位寻址格式

按位寻址时的格式为 AX.Y，使用时必须指定元件名称、字节地址和位号，如图 2-7 所示是输入继电器（I）的位寻址格式举例。

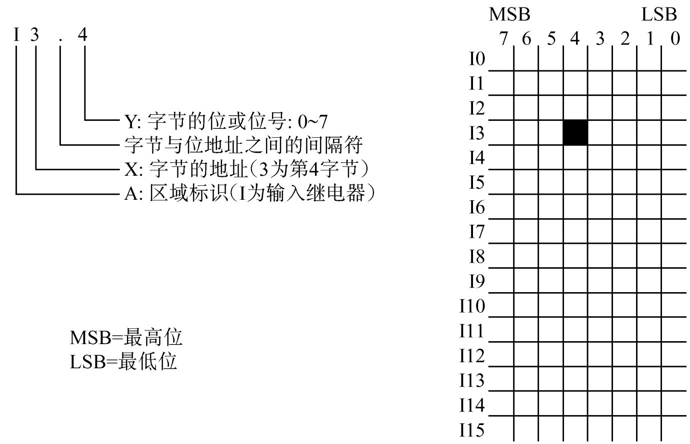


图 2-7　CPU 存储器中位数据表示方法举例（位寻址）

可以进行这种方式位寻址的编程元件有：输入继电器（I）、输出继电器（Q）、

通用辅助继电器（M）、特殊继电器（SM）、局部变量存储器（L）、变量存储器（V）和顺序控制继电器（S）。

3. 特殊器件的寻址格式

存储区内另有一些元件是具有一定功能的器件，不用指出它们的字节，而是直接写出其编号。这类元件包括定时器（T）、计数器（C）、高速计数器（HSC）和累加器（AC）。其中 T 和 C 的地址编号中均包含两个含义，如 T10，既表示 T10 的定时器位状态信息，又表示该定时器的当前值。

累加器（AC）的数据长度可以是字节、字或双字。使用时只表示出累加器的地址编号，如 AC0，数据长度取决于进出 AC0 的数据类型。

4. 字节、字和双字的寻址格式

对字节、字和双字数据，直接寻址时需指明元件名称、数据类型和存储区域内的首字节地址。如图 2-8 所示是以变量存储器（V）为例分别存取 3 种长度数据的比较。可以用此方式进行寻址的元件有输入继电器（I）、输出继电器（Q）、通用辅助继电器（M）、特殊继电器（SM）、局部变量存储器（L）、变量存储器（V）、顺序控制继电器（S）、模拟量输入映像寄存器（AI）和模拟量输出映像寄存器（AQ）。

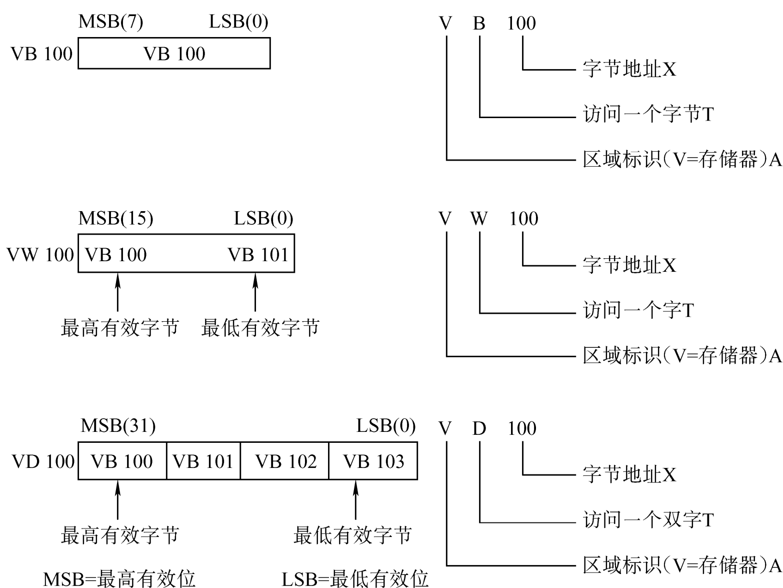


图 2-8　存取 3 种长度数据的比较

四、CPU 存储区域（软元件）的间接寻址

在直接寻址方式中，直接使用存储器或寄存器的元件名称和地址编号，根据这个地址可以立即找到该数据。

间接寻址方式是指数据存放在存储器或寄存器中，在指令中只出现所需数据所在单元的内存地址的地址。存储单位地址的地址又称为地址指针。这种间接寻址方式与计算机的间接寻址方式相同。间接寻址在处理内存连续地址中的数据时非常方便，而且可以缩短程序所生成的代码长度，使编程更加灵活。

可以用指针进行间接寻址的存储区有输入继电器（I）、输出继电器（Q）、通用辅助继电器（M）、变量存储器（V）、顺序控制继电器（S）、定时器（T）和计数器（C）。其中 T 和 C 仅仅是当前值可以进行间接寻址，而对独立的位值和模拟量值不能进行间接寻址。

使用间接寻址方式存取数据方法与 C 语言中的指针应用基本相同，其过程如下。

1. 建立指针

使用间接寻址对某个存储器单元读、写时，首先要建立地址指针。指针为双字长，是所要访问的存储单元的 32 位的物理地址，可作为指针的存储区有变量存储器（V）、局部变量存储器（L）和累加器（AC1、AC2、AC3）。必须用双字传送指令（MOVD），将所要访问的存储器单元的地址装入用来作为指针的存储器单元或寄存器，装入的是地址而不是数据本身。格式如下：

例：MOVD　&VB100，VD204
　　MOVD　&VB10，AC2
　　MOVD　&C2，LD16

其中：“&”为地址符号，它与单元编号结合使用表示所对应单元的 32 位物理地址；VB100 只是个直接地址编号，并不是它的物理地址。指令中的第二个地址数据长度必须是双字长，如：VD、LD 和 AC 等。

2. 用指针来存取数据

在操作数的前面加“*”表示该操作数为一个指针。如图 2-9 所示，AC1 为指针，用来存放要访问的操作数的地址。在这个例子中，存于 VB200、VB201 中的数据被传送到 AC0 中去。

3. 修改指针

连续存储数据时，可以通过修改指针很容易存取其紧接的数据。简单的数学运算指令，如加法、减法、自增和自减等指令可以用来修改指针。在修改指针时，要记住访问数据的长度：存取字节时，指针加 1；存取字时，指针加 2；存取双字时，指针加 4。图 2-9 说明如何建立指针，如何存取数据及修改指针。

五、软元件及操作数的寻址范围

S7-200 PLC 的 CPU22X 系列的编程元件的寻址范围见表 2-12。

S7-200 PLC 的 CPU22X 系列指令操作数的有效寻址范围见表 2-13。

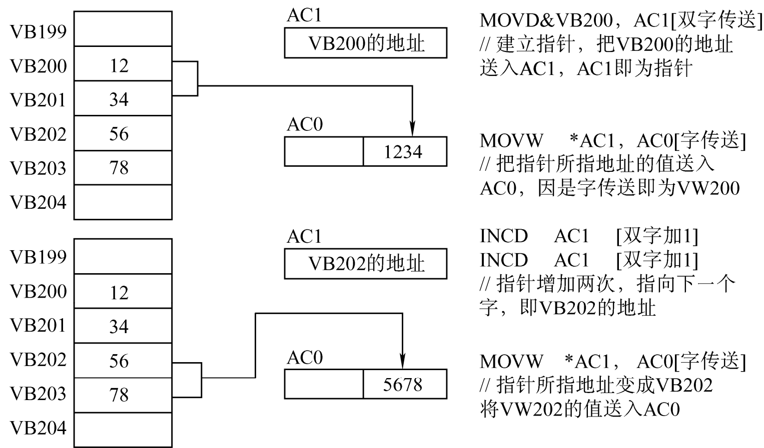


图 2-9　指针的建立、存取及修改

表 2-12　S7-200 PLC 的 CPU22X 系列的编程元件的寻址范围

编程元件	CPU 221	CPU 222	CPU 224	CPU 226
用户程序	2KB		4KB	
用户数据	1KB		2.5KB	
输入继电器（I）	I0.0～I15.7			
输出继电器（Q）	Q0.0～Q15.7			
模拟量输入映像寄存器（AIW）	AIW0～AIW30			
模拟量输出映像寄存器（AQW）	AQW0～AQW30			
变量寄存器（V）	VB0.0～VB2047.7		VB0.0～VB5119.7	
局部变量寄存器（L）	LB0.0～LB63.7			
辅助继电器（M）	M0.0～M31.7			
特殊继电器（SM） 只读特殊继电器（SM）	SM0.0～SM299.7 SM0.0～SM29.7			
定时器（T）	T0～T255			
计数器（C）	C0～C255			
高速计数器（HSC）	HC0，HC3，HC4，HC5		HC0～HC5	
状态继电器（S）	S0.0～S31.7			
累加器（AC）	AC0～AC3			
跳转标号	0～255			
调用子程序	0～63			
中断程序	0～127			
PID 回路	0～7			
通信口	0	0	0	0，1

表 2-13　S7-200 PLC 的 CPU22X 系列指令操作数的有效寻址范围

操作数类型	CPU221	CPU222	CPU224，CPU226
位	I0.0～15.7，Q0.0～15.7 M0.0～31.7，S0.0～31.7 SM0.0～179.7，T0～255 V0.0～2047.7，C0～255 L0.0～63.7	I0.0～15.7，Q0.0～15.7 M0.0～31.7，S0.0～31.7 SM0.0～179.7，T0～255 V0.0～2047.7，C0～255 L0.0～63.7	I0.0～15.7，Q0.0～15.7 M0.0～31.7，S0.0～31.7 SM0.0～179.7，T0～255 V0.0～5119.7，C0～255 L0.0～63.7
字　节	IB0～15，QB0～15 MB0～31，SM0～179 SB0～31，VB0～2047 LB0～63，AC0～3 常数	IB0～15，QB0～15 MB0～31，SMB0～179 SB0～31，VB0～2047 LB0～63，AC0～3 常数	IB0～15，QB0～15 MB0～31，SMB0～179 SB0～31，VB0～5119 LB0～63，AC0～3 常数
字	IW0～14，QW0～14 MW0～30，SMW0～178 SW0～30，VW0～2046 LW0～62，AC0～3 T0～255，C0～255 常数	IW0～14，QW0～14 MW0～30，SMW0～178 SW0～30，VW0～2046 LW0～62，AC0～3 T0～255，C0～255 AIW0～30，AQW0～30 常数	IW0～14，QW0～14 MW0～30，SMW0～178 SW0～30，VW0～5118 LW0～62，AC0～3 T0～255，C0～255 AIW0～25，AQW0～30 常数
双　字	ID0～12，QD0～12 MD0～28，SMD0～176 SD0～28，VD0～2044 LD0～60，AC0～3 HC0，HC3～5，常数	ID0～12，QD0～12 MD0～28，SMD0～176 SD0～28，VD0～2044 LD0～60，AC0～3 HC0，HC3～5，常数	ID0～12，QD0～12 MD0～28，SMD0～176 SD0～28，VD0～5116 LD0～60，AC0～3 HC0～5，常数

第四节　S7-200 PLC 的编程语言和程序结构

一、S7-200 PLC 的编程语言

与个人计算机相比，PLC 的硬件、软件的体系结构都是封闭的而不是开放的。各厂家的 PLC 编程语言和指令系统的功能和表达方式也不一致，有的甚至有相当大的差异，因此各厂家的 PLC 互不兼容。IEC（国际电工委员会）是为电工电子技术的所有领域制订标准的世界性组织。IEC 于 1994 年 5 月公布了 PLC 标准（IEC 61131），该标准鼓励不同的 PLC 制造商提供在外观和操作上相似的指令。它由以下 5 部分组成：通用信息、设备与测试要求、编程语言、用户指南和通信。其中的第三部分（IEC 61131-3）是 PLC 的编程语言标准。IEC 61131-3 标准使用户在使用新的 PLC 时，可以减少重新培训的时间；对于厂家，使用标准将减少产品开发的时间，可以投入更多的精力去满足用户的特殊要求。

目前，已有越来越多的生产 PLC 的厂家提供符合 IEC 61131-3 标准的产品，有的厂家推出的在个人计算机上运行的“软件 PLC”软件包也是按 IEC 61131-3 标准设计的。

IEC 61131-3 详细地说明了句法、语义和下述 5 种编程语言的表达方式如图 2-10 所示：

① 顺序功能图（Sequential Function Chart，SFC）；

② 梯形图（Ladder Diagram，LD）；

③ 功能块图（Function Block Diagram，FBD）；

④ 指令表（Instruction List，IL）；

⑤ 结构文本（Structured Text，ST）。



图 2-10　PLC 的编程语言

标准中有两种图形语言——梯形图（LD）和功能块图（FBD），还有两种文字语言——指令表（IL）和结构文本（ST）。可以认为，顺序功能图（SFC）是一种结构块控制程序流程图。

1. 顺序功能图（SFC）

这是一种位于其他编程语言之上的图形语言，用来编制顺序控制程序，顺序功能图提供了一种组织程序的图形方法，在顺序功能图中可以用别的语言嵌套编程。步、转换和动作是顺序功能图中的 3 种主要元件。可以用顺序功能图来描述系统的功能，根据它可以很容易地画出梯形图程序。

2. 梯形图（LD）

梯形图是使用得最多的 PLC 图形编程语言。梯形图与继电器控制系统的电路图很相似，具有直观、易懂的优点，很容易被工厂熟悉继电器控制的电气人员掌握，特别适用于数字量逻辑控制。有时把梯形图称为电路或程序。

梯形图由触点、线圈和用方框表示的功能块组成。触点代表逻辑输入条件，例如外部的开关、按钮和内部条件等。线圈通常代表逻辑输出结果，用来控制外部的指示灯、交流接触器和内部的输出条件等。功能块用来表示定时器、计数器或者数学运算等附加指令。

在分析梯形图中的逻辑关系时，为了借用继电器电路图的分析方法，可以想象左右两侧垂直“电源线”之间有一个左正右负的直流电源电压（S7-200 PLC 的梯形图中省略了右侧的垂直电源线），当图 2-11 中的 I0.1 与 I0.2 的触点接通，或 M0.3 与 I0.2 的触点接通时，有一个假想的“能流”（Power Flow）流过 Q1.1 的线

圈。利用能流这一概念，可以帮助我们更好地理解和分析梯形图，能流只能从左向右流动。

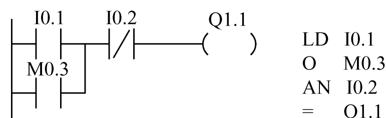


图 2-11　梯形图与语句表

触点和线圈等组成的独立电路称为网络（Network），用编程软件生成的梯形图和语句表程序中有网络编号，允许以网络为单位，给梯形图加注释。本书为节约篇幅，一般没有标注网络号。在网络中，程序的逻辑运算按从左到右的方向执行，与能流的方向一致。各网络按从上到下的顺序执行，执行完所有的网络后，返回最上面的网络重新执行。

使用编程软件可以直接生成和编辑梯形图，并将它下载到 PLC。

3. 功能块图（FBD）

这是一种类似于数字逻辑门电路的编程语言，有数字电路基础的人很容易掌握。该编程语言用类似与门、或门的方框来表示逻辑关系，方框的左侧为逻辑运算的输入变量，右侧为输出变量，输入、输出端的小圆圈表示“非”运算，方框被“导线”连接在一起，信号自左向右流动。图 2-12 中的控制逻辑与图 2-11 中相同。西门子公司的“LOGO！”系列微型 PLC 使用功能块图语言，除此之外，国内很少有人使用功能块图语言。

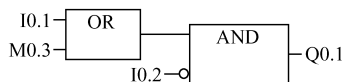


图 2-12　功能块图

4. 语句表（STL）

S7 系列 PLC 将指令表称为语句表。PLC 的指令表是一种与微机的汇编语言中的指令相似的助记符表达式，由指令组成的程序叫做指令表程序或语句表程序。

语句表比较适合熟悉 PLC 和逻辑程序设计的经验丰富的程序员使用。

5. 结构文本（ST）

结构文本（ST）是为 IEC 61131-3 标准创建的一种专用的高级编程语言。与梯形图相比，它能实现复杂的数学运算，编写的程序非常简洁和紧凑。

6. 编程语言的相互转换和选用

在 S7-200 PLC 的编程软件中，用户可以选用梯形图、功能块图和语句表这 3 种编程语言。语句表不使用网络，但是可以用 Network（网络）这个关键词对程序分段，这样的程序可以转换为梯形图。

语句表程序较难阅读，其中的逻辑关系很难一眼看出，所以在设计复杂的数字量控制程序时一般使用梯形图语言。语句表可以处理某些不能用梯形图处理的问题，梯形图编写的程序一定能转换为语句表。

梯形图程序中输入信号与输出信号之间的逻辑关系一目了然，易于理解，与继电器电路图的表达方式极为相似，设计数字量控制程序时建议选用梯形图语言。语句表输入方便快捷，梯形图中功能块对应的语句只占一行的位置，还可以为每一条语句加上注释，便于复杂程序的阅读。在设计通信、数学运算等高级应用程

序时建议使用语句表语言。

7. SIMATIC 指令集与 IEC 61131-3 指令集

供 S7-200 PLC 使用的 STEP 7-Micro/WIN 编程软件提供两种指令集：SIMATIC 指令集与 IEC 61131-3 指令集。前者由西门子公司提供，它的某些指令不是 IEC 61131-3 中的标准指令。通常 SIMATIC 指令的执行时间短，可以使用梯形图、功能块图和语句表语言，而 IEC 61131-3 指令集只提供前两种语言。

IEC 61131-3 指令集的指令较少，其中的某些“块”指令可以接受多种数据格式。例如 SIMATIC 指令集的加法指令被分为 ADD_I（整数加）、ADD_DI（双字整数加）与 ADD_R（实数加）等，IEC 61131-3 的加法指令 ADD 则未做区分，而是通过检验数据格式，由 CPU 自动选择正确的指令。IEC 指令通过检查参数中的数据格式错误还可以减少程序设计中的错误。

在 IEC 61131-3 指令编辑器中，有些是 SIMATIC 指令集中的指令，它们作为 IEC 61131-3 指令集的非标准扩展，在编程软件的指令树内用红色的“+”号标记。

二、S7-200 PLC 的程序结构

一个系统的控制功能是由用户程序决定的。为完成特定的控制任务，需要编写用户程序，使得 PLC 能以循环扫描的工作方式执行用户程序。在 SIMATIC S7 系列中，为适应用户程序的不同需求，STEP7 为用户提供了 3 种程序设计方法，其程序结构分别为：线性化编程、分部式编程和结构化编程。

1. 程序结构

（1）线性化编程　所谓线性化编程就是将用户程序连续放置在一个指令块内，这个指令块在 SIMATIC 的 PLC 中，通常称为组织块（OB1）。CPU 周期性地扫描 OB1，使用户程序在 OB1 内顺序执行每条指令。

由于线性化编程将全部指令都放在一个指令块中，它的程序结构具有简单、直接的特点，适合由一个人编写用户程序。S7-200 PLC 就是采用线性化编程方法。

（2）分部式编程　所谓分部式编程就是将一项控制任务分成若干个指令块，每个指令块用于控制一套设备或者完成一部分工作。每个指令块的工作内容与其他指令块的工作内容无关，一般没有子程序的调用，这些指令块的运行是通过组织块（OB1）内的指令来调用。例如，一个分部式程序可能包含以下指令块：

① 用于控制设备每一部分的功能块 FC（如 FC10）；

② 用于控制设备每一工作状态的功能块 FC（如 FC20）；

③ 用于控制操作员接口的功能块 FC（如 FC30）；

④ 用于处理诊断逻辑的功能块 FC（如 FC40）。

在分部式程序中，既无数据交换也无重复利用的代码，因此分部式编程允许多个设计人员同时编写用户程序，而不会发生内容冲突。分部式编程的程序调用

结构如图 2-13 所示。

（3）结构化编程　所谓结构化编程是将整个用户程序分成一些具有独立功能的指令块，其中有若干个子程序块，然后再按照要求调用各个独立的指令块，从而构成一个完整的用户程序。结构化编程的特点：编程简单，结构清晰，可以采用子程序技术使部分程序标准化，调试方便。一般比较大型的控制程序，均采用结构化编程。结构化编程的程序调用结构如图 2-14 所示。

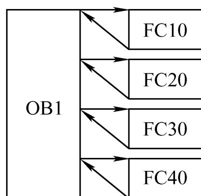


图 2-13　分部式编程的程序调用结构

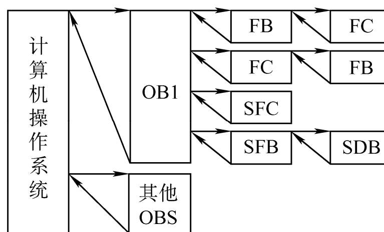


图 2-14　结构化编程的程序调用结构

2. S7-200 PLC 的程序结构

S7-200 PLC 的程序结构属于线性化编程，其用户程序一般由 3 部分构成：用户程序、数据块和参数块。

（1）用户程序　一个完整的用户程序一般是由一个主程序、若干子程序和若干个中断处理子程序组成的。对线性化编程，主程序应安排在程序的最前面，其次为子程序和中断程序，如图 2-15 所示。

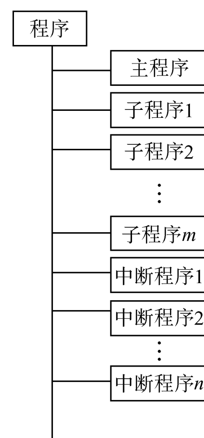


图 2-15　S7-200 PLC 的程序结构

如果用工业编程软件 STEP 7-Micro/ WIN32 在计算机上编程，可以用两种方法组织程序结构，一种方法是利用编程软件的程序结构窗口，分别双击主程序、子程序和中断程序的图标，即可进入各个程序块的编程窗口。编译时编程软件自动对各个程序段进行连接。另一种方法是只进入主程序窗口，将主程序、子程序和中断程序按顺序依次安排在主程序窗口。

（2）数据块　S7-200 PLC 中的数据块，一般为 DB1，主要用来存放用户程序运行所需的数据。在数据块中允许存放的数据类型为：布尔型、十进制、二进制、十六进制、字母、数字和字符型。

（3）参数块　在 S7-200 PLC 中，参数块中存放的是 CPU 组态数据，如果在编程软件或其他编程工具上未进行 CPU 的组态，则系统以默认值进行自动配置。

（4）主程序　主程序（OB1）是程序的主体，每一个项目都必须并且只能有一个主程序。在主程序中可以调用子程序和中断程序。

主程序通过指令控制整个应用程序的执行，每次 CPU 扫描都要执行一次主程序。STEP 7-Micro/WIN 的程序编辑器窗口下部的标签用来选择不同的程序。因为各个程序已被分开，各程序结束时不需要加入无条件结束指令或无条件返回指令。

（5）子程序　子程序是一个可选的指令的集合，仅在被其他程序调用时执行。同一子程序可以在不同的地方被多次调用，使用子程序可以简化程序代码和减少扫描时间。设计得好的子程序容易移植到别的项目中去。

（6）中断程序　中断程序是指令的一个可选集合，中断程序不是被主程序调用，它们在中断事件发生时由 PLC 的操作系统调用。中断程序用来处理预先规定的中断事件，因为不能预知何时会出现中断事件，所以不允许中断程序改写可能在其他程序中使用的存储器。

第三章　S7-200 PLC 基本指令与应用指令

在 S7-200 PLC 的指令系统中，可分为基本指令与应用指令。最初把能够取代传统的继电器控制系统的那些指令称为基本指令，而应用指令是指为满足用户不断提出的一些特殊控制要求开发出的那些指令，应用指令又称为功能指令。由于 PLC 的功能越来越强，涉及的指令也越来越多，对基本指令所包含的内容也在不断扩充，所以，基本指令与应用指令目前还没有严格的界限与区分。在本章中，我们将由简到繁介绍一些在实际工程中经常用到的 S7-200 PLC 的指令。

第一节　位操作指令

PLC 的位操作指令主要实现逻辑控制和顺序控制，完全可以用 S7-200 PLC 的位操作指令代替传统的继电器/接触器控制系统。

一、基本逻辑指令

1. 触点指令

触点及线圈指令是 PLC 中应用最多的指令。触点首先分为动合触点及动断触点，又以其在梯形图中的位置分为和母线相连的动合触点或动断触点、与前边触点串联的动合或动断触点、并联的动合或动断触点。一些型号 PLC 还有边沿脉冲触点指令及取反触点指令。边沿脉冲触点指令是在满足工作条件时，接通一个扫描周期，取反触点指令是将送入的能流取反后送出。表 3-1 为西门子 S7-200 PLC 的触点指令。

表 3-1　S7-200 PLC 部分触点指令表

指令			梯形图符号	数据类型	操作数	指令功能
标准触点	动合	LD	Bit	位	I、Q、V、M、SM、S、T、C	将动合触点接在母线上
		A	Bit			动合触点与其他程序段相串联
		O	Bit			动合触点与其他程序段相并联
	动断	LDN	Bit			将动断触点接在母线上

（续）

指令			梯形图符号	数据类型	操作数	指令功能
标准触点	动断	AN	Bit ─┤/├─	位	I、Q、V、M、SM、S、T、C	动断触点与其他程序段相串联
		ON	Bit ─┤/├─ (并联)			动断触点与其他程序段相并联
取反		NOT	─┤NOT├─		—	改变能流输入状态
正负跳变	正	EU	─┤P├─		—	检测到一次正跳变，能流接通一个扫描周期
	负	ED	─┤N├─		—	检测到一次负跳变，能流接通一个扫描周期

2. 线圈指令

线圈指令用来表达一段程序的运算结果。线圈指令含普通线圈指令、置位及复位线圈指令、立即线圈指令等类型。普通线圈指令在工作条件满足时，将该线圈相关存储器置 1，在工作条件失去后复零。置位线圈指令在相关工作条件满足时将有关线圈置 1，工作条件失去后，这些线圈仍保持置 1，复位需用复位线圈指令。立即线圈指令采用中断方式工作，可以不受扫描周期的影响，将程序运算的结果立即送到输出口。表 3-2 为西门子 S7-200 PLC 的线圈指令。

表 3-2　S7-200 PLC 线圈指令表

指令与助记符		梯形图符号	数据类型	操作数	指令功能
输出	=	Bit ─( )	位	Q、V、M、SM、S、T、C	将运算结果输出到某个继电器
立即输出	=I	Bit ─(I)	位	Q	立即将运算结果输出到某个继电器
置位与复位	S	Bit ─(S) N	位 N：BYTE 或常数	位：Q、V、M、SM、S、T、C N：IB、QB、VB、SMB、SB、LB、AC、MB、常数等	将从指定地址开始 N 个位置位
	R	Bit ─(R) N	位 N：BYTE 或常数	位：Q、V、M、SM、S、T、C N：IB、QB、VB、SMB、SB、LB、AC、MB、常数等	将从指定地址开始 N 个位复位
立即置位与立即复位	SI	Bit ─(SI) N	位 N：BYTE 或常数	位：Q N：IB、QB、VB、SMB、SB、LB、AC、MB、常数等	立即将从指定地址开始 N 个位置位
	RI	Bit ─(RI) N	位 N：BYTE 或常数	位：Q N：IB、QB、VB、SMB、SB、LB、AC、MB、常数等	立即将从指定地址开始 N 个位复位

（续）

指令与助记符		梯形图符号	数据类型	操 作 数	指 令 功 能
SR 触发器	SR	Bit SI OUT SR R	位	Q、V、M、I、S	置位与复位同时为 1 时置位优先
RS 触发器	RS	Bit SI OUT RS RI	位	Q、V、M、I、S	置位与复位同时为 1 时复位优先

3. 触点及线圈指令梯形图实例

过去接触过的继电器/接触器控制系统中，控制电动机的起停往往需要两只按钮，在这里我们利用 PLC 逐行扫描的特点使用一只按钮控制电动机的起停，实现这个控制目的的方案很多，下面是其中 3 个方案。1 个例子用了 3 个方案的目的是一方面再熟悉一下周期性扫描的特点，另一方面是说明编程的灵活性。

将起动/停止的输入信号接按钮的常开触点并连接到输入点 I0.0，并通过输出点 Q1.0 连接接触器线圈来控制电动机。操作方法是：按一下该按钮，输入的是起动信号，再按一下该按钮，输入的则是停止信号，以此形成单数次时为起动，双数次时为停止，方案 1 如图 3-1 所示。

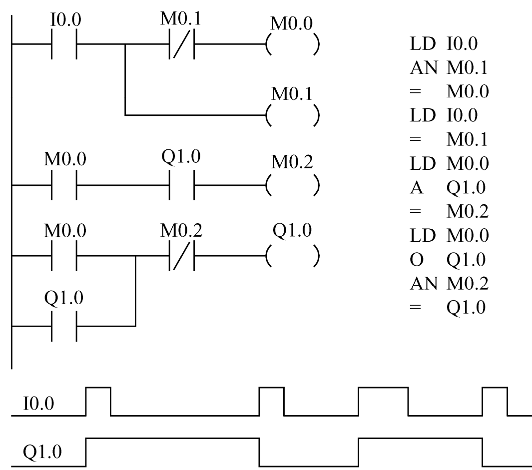


图 3-1 单按钮控制电动机起、停控制方案 1

当第 1 次按下按钮时，在当前扫描周期内，I0.0 使辅助继电器 M0.0，M0.1 为 ON 状态，使 Q1.0 为 ON；到第二个扫描周期，辅助继电器 M0.1 的动断触点为 OFF，使 M0.0 为 OFF，辅助继电器 M0.2 仍为 OFF，M0.2 的动断触点仍为 ON，Q1.0 的自锁触点已起作用，Q1.0 仍为 ON，从此不管经过多少扫描周期，这种状

态也不会改变。第 1 次松开按钮后至第 2 次按下按钮前，在输入采样阶段读入 I0.0 的状态为 OFF，辅助继电器 M0.0、M0.1、M0.2 均为 OFF 状态，Q1.0 也继续保持 ON 状态。当第二次按下按钮时，在当前扫描周期时，辅助继电器 M0.0、M0.1、M0.2 均为 ON 状态，M0.2 的动断触点为 OFF，使 Q1.0 由 ON 变为 OFF；到下一个扫描周期（假定末松开按钮），M0.1 的动断触点使 M0.0 为 OFF，使 M0.2 为 OFF，Q1.0 不具备吸合条件仍然为 OFF。第 2 次松开按钮后至第 3 次按下按钮前，M0.0、M0.1、M0.2 及 Q1.0 均为 OFF 状态，控制程序恢复为原始状态。所以，当第 3 次按下按钮时，又开始了起动操作，由此进行起、停电动机。

方案 2 如图 3-2 所示。相对于方案 1 去掉一个中间环节，增加了一个正跳变指令，这个指令的特点就是当处在其前面的触点信号从 OFF 变 ON 时，它只 ON 一个扫描周期。当按一下按钮时，I0.0 由 OFF 变 ON，这时上升沿（正跳变）触发 EU 指令使 M0.0 只 ON 一个扫描周期，在本周期内接下来的扫描行是定 M0.1 的状态，因 M0.0 是 ON，而 Q1.0 是 OFF，所以 M0.1 是 OFF。最后是定 Q1.0 的状态，因 M0.0 是 ON，而 M0.1 是 OFF，那 M0.1 的动断触点是 ON，这样使得 Q1.0“得电吸合”成为 ON 状态，接在这一点上的控制电动机的接触器线圈便得电吸合，电动机就可转动起来。在接下来的第二扫描周期，即使按钮还没有松开，I0.0 还处于 ON 状态，由于 P 指令的作用，M0.0 变成了 OFF，也就是说从第二周期开始 M0.0 总是 OFF 了，下面的 M0.1 也不具备“得电吸合”的条件始终处于 OFF 状态，Q1.0 仍然是 ON 状态。接下来就是松开按钮，三个线圈的状态仍然与第二扫描周期的相同，电动机也始终在转动着。当我们第二次按下按钮时，就会形成 M0.0 与 M0.1 都是 ON 状态，而 Q1.0 成为 OFF 状态，电动机便停止转动了。从第二次按下按钮的第二扫描周期开始三个线圈的状态都变成 OFF 了，恢复为原始状态。在这以后，当第 3 次按下按钮时，又开始了起动操作，由此进行起、停电动机。

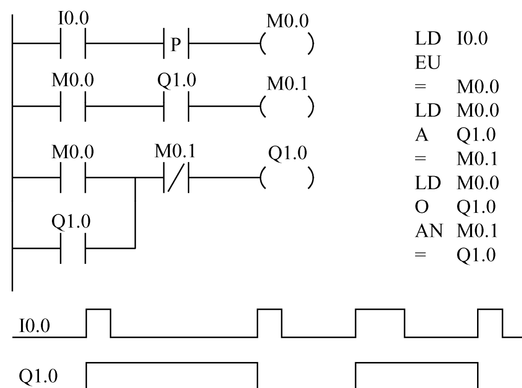


图 3-2　单按钮控制电动机起停控制方案 2

方案 3 如图 3-3 所示。在这里使用了 RS 触发器及上升沿（正跳变）触发 EU（P）指令，利用 P 指令只 ON 一个扫描周期的特点以及 RS 触发器在置位与复位同时为 1 时复位信号优先的特点，实现单按钮控制电动机起、停的目的。当第一次按下按钮时，在当前扫描周期，I0.0 成为 ON 状态，RS 触发器的置位端为 1，而复位端由于 Q1.0 此时处于 OFF 状态使得复位端为 0，所以在第一次按下按钮的第一个扫描周期，Q1.0 就会成为 ON 状态，电动机起动运行。从第二个扫描周期开始，由于 P 指令的作用，RS 触发器的置位与复位端都为 0，Q1.0 继续保持 ON 状态，无论继续按住或松开按钮，这样的状态也不会改变了。当我们第二次按下按钮时，由于 Q1.0 已经是 ON 状态了，所以就会形成触发器的置位端与复位端都为 1 的时刻，这样由于 RS 触发器是复位优先，就会使得 Q1.0 复位，变成 OFF 状态，电动机就停止运行了。同样这种方案也能形成单数次按下按钮时为起动，双数次时为停止。

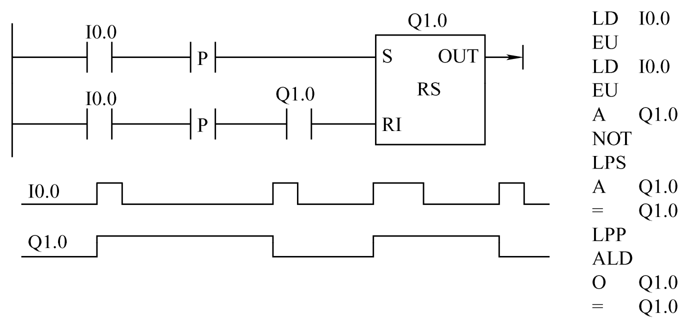


图 3-3 单按钮控制电动机起、停控制方案 3

二、定时器指令

1. 指令

定时器是 PLC 中最常用器件之一，准确用好定时器对于 PLC 程序设计非常重要。S7-200 PLC 的 CPU22X 系列 PLC 的定时器有 3 种类型：接通延时型 TON；保持型（有记忆的）接通延时型 TONR；断开延时型 TOF。

定时器指令用来规定定时器的功能，表 3-3 为西门子 S7-200 系列 PLC 定时器指令表，3 条指令规定了三种不同功能的定时器。

西门子 S7-200 系列定时器使用的基本要素如下：

（1）编号、类型及精度 S7-200 系列 PLC 配置了 256 个定时器，编号为 T0～T255。定时器有 1ms、10ms、100ms 三种精度，1ms 的定时器有 4 个，10ms 的定时器有 16 个，100ms 的定时器有 236 个。编号和类型与精度有关，例如编号是 T2 的精度是 10ms，类型为有记忆的接通延时型。选用前应先查表 3-4 以确定合适

的编号，从表中可知，有记忆的定时器均是接通延时型，无记忆的定时器可通过指令指定为接通延时或断开延时型，使用时还须注意，在一个程序中不能把一个定时器同时用做不同类型，如既有 TON37 又有 TOF37。

表 3-3　定时器指令类别表

定时器类别	接通延时定时器	保持型接通延时定时器	断开延时定时器
指令的表达形式	T×× IN TON PT ××ms	T×× IN TONR PT ××ms	T×× IN TOF PT ××ms
操作数的范围及类型	T××：字型；常数 T0～T255，指定定时器号 IN：位型；I、Q、V、M、SM、S、T、C、L、能流，起动定时器 PT：整数型；IW、QW、VW、MW、SMW、T、C、LW、AC、AIW、*VD、*LD、*AC、常数，设定值输入端		

注：带“*”的存储单元具有变址功能。

表 3-4　定时器的精度及编号

定时器类型	定时精度/ms	最大当前值/s	定时器编号
TONR（有记忆）	1	32.767	T0，T64
	10	327.67	T1～T4，T65～T68
	100	3276.7	T5～T31，T69～T95
TON，TOF（无记忆）	1	32.767	T32，T96
	10	327.67	T33～T36，T97～T100
	100	3276.7	T37～T63，T101～T255

（2）预置值（也叫设定值）　预置值即编程时设定的延时时间的长短，PLC 定时器采用时基计数及与预置值比较的方式确定延时时间是否达到，时基计数值称为当前值，存储在当前值寄存器中，预置值在使用梯形图编程时，标在定时器功能框的“PT”端。

（3）工作条件　工作条件也叫使能输入，从梯形图的角度看，定时器功能框中“IN”端连接的是定时器的工作条件。对于接通延时型定时器来说，有能流流到“IN”端时开始计时；对于断开延时型定时器来说，能流从有变到无时开始计时；对于无记忆的定时器来说，工作条件失去，如接通延时型定时器能流从有变到无时，无论定时器计时是否达到预置值，定时器均复位，前边的计时值清零；对于有记忆定时器来说，可累计分段的计时时间，这种定时器的复位就得靠复位指令了。

（4）工作对象　工作对象指定时时间到时，利用定时器的触点控制的元器件或工作过程。S7-200 系列 PLC 定时器的工作过程可以描述如下：

接通延时定时器和有记忆的接通延时定时器在“IN”端接通，定时器的当前值大于等于 PT 端的预置值时，该定时器位被置位。当达到预设时间后，接通延时定时器和有记忆的接通延时定时器继续计时，后者的当前值可以分段累加，一直到最大值 32767。

断开延时定时器在使能输入“IN”接通时，定时器位立即接通，并把当前值设为 0。当“IN”端断开时起动计时，达到预设值 PT 时，定时器位断开，并且停止当前值计数。当“IN”端断开的时间短于预置值时，定时器位保持接通。

（5）S7-200 的定时器的刷新方式　S7-200 PLC 的定时器有 3 种不同的定时精度，即每种定时精度对应不同的时基脉冲。定时器计时的过程就是数时基脉冲的过程。然而，这 3 种不同定时精度的定时器的刷新方式是不同的，要正确使用定时器，首先要知道定时器的刷新方式，保证定时器在每个扫描周期都能刷新 1 次，并能执行 1 次定时器指令。

① 1ms 定时器的刷新方式　1ms 定时器采用中断刷新的方式，系统每隔 1ms 刷新 1 次，与扫描周期即程序处理无关。当扫描周期较长时，1ms 的定时器在 1 个扫描周期内将多次被刷新，其当前值在每个扫描周期内可能不一致。

② 10ms 定时器的刷新方式　10ms 的定时器由系统在每个扫描周期开始时自动刷新，在每次程序处理阶段，定时器位和当前值在整个扫描过程中不变。在每个扫描周期开始时将一个扫描周期累计的时间加到定时器当前值上，例：扫描周期是 30ms 的程序，这个定时器在“IN”端接通有效到本周期结束用时 18ms，下个周期整个扫描过程中的当前值都是 18ms，再下个周期就是 48ms，再下个周期就是 78ms，假设我们的定时器的预置值是 70ms，在这个周期，定时器的位就可起作用了，实际计时超过 70ms。

③ 100ms 定时器的刷新方式　100ms 的定时器是在该定时器指令执行时被刷新。为了使定时器正确地定时，要确保每个扫描周期都能执行一次 100ms 定时器指令，程序的长短会影响定时的准确性。

④ 正确使用定时器　在 PLC 的应用中，经常使用具有自复位功能的定时器，即利用定时器自己的动断触点去控制自己的线圈。在 S7-200 PLC 中，要使用具有自复位功能的定时器，必须考虑定时器的刷新方式。

图 3-4a 中，T96 是 1ms 的定时器，只有正好在程序扫描到 T96 的动断触点到 T96 的动合触点之间当前值等于预置值时被刷新，进行状态位的转换，使 T96 的动合触点为 ON，从而使 M0.0 能 ON 一个扫描周期，否则 M0.0 将总是 OFF 状态。正确解决这个问题的方法是采用图 3-4b 的编程方式。

图 3-5a 中，T33 是 10ms 的定时器，而 10ms 的定时器是在扫描周期开始时被刷新的，由于 T33 的动断触点和动合触点的相互矛盾状态，使得 M0.0 永远为 OFF 状态。正确解决这个问题的方法是采用图 3-5b 的编程方式。

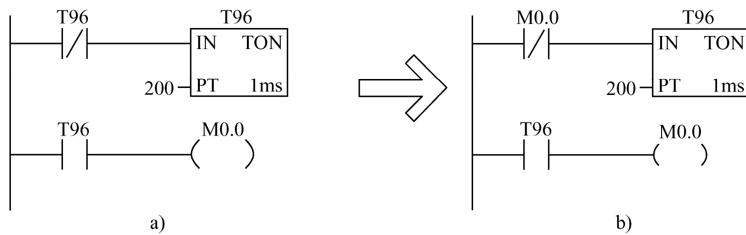


图 3-4　1ms 定时器的正确使用

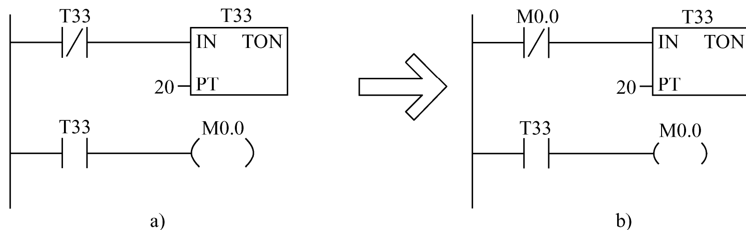


图 3-5　10ms 定时器的正确使用

对于 100ms 的定时器，推荐采用图 3-6b 编程方式。

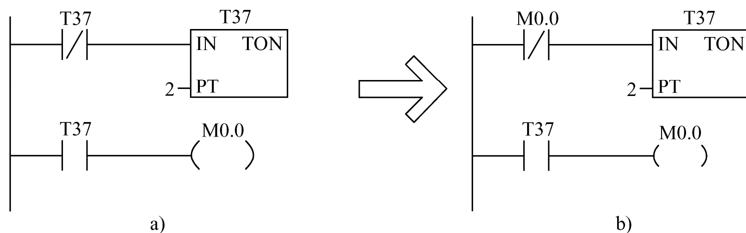


图 3-6　100ms 定时器的正确使用

2. 定时器应用举例

用定时器设计输出脉冲的周期和占空比可调的振荡电路（即闪烁电路）。

在图 3-7 中，在 I0.0 处于 OFF 状态时，T37 与 T38 也都处于 OFF 状态。当 I0.0 处于 ON 状态后，T37 的 IN 输入端为 1 状态，T37 开始定时。3s 后定时时间到，T37 的动合触点接通，使 Q1.0 变为 ON，同时 T38 开始定时。5s 后定时时间到，它的动断触点断开，使 T37 的 IN 输入端变为 0 状态，T37 的动合触点断开，使 Q1.0 变为 OFF，同时 T38 因为 IN 输入端变为 0 状态，它被复位。复位后其动断触点又接通，T37 又开始计时，往后 Q1.0 的线圈就这样周期性地“通电”与“断电”，直到 I0.0 变为 OFF，Q1.0 线圈“通电”与“断电”的时间分别等于 T38 与 T37 的预置值。闪烁电路实际上是一个具有正反馈的振荡电路，T37 与 T38 的输出信号通过它们的触点分别控制对方的线圈，形成了正反馈。另外，特殊继电器

SM0.5 是以触点形式供我们使用的，它可提供周期为 1s，占空比为 0.5 的脉冲信号，此脉冲信号是不可调的，利用它也可以驱动需要闪烁的指示灯。

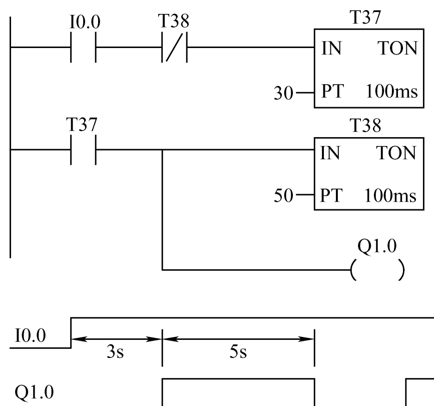


图 3-7　振荡电路梯形图及时序图

三、计数器指令

S7-200 PLC 的普通计数器有 3 种类型：递增计数器 CTU、递减计数器 CTD 和增减计数器 CTUD，共计 256 个，编号为 C0～C255。可根据实际编程需要，对某个计数器的类型进行定义。不能重复使用同一个计数器的线圈编号，即每个计数器的线圈编号只能使用 1 次。每个计数器有一个 16 位的当前值寄存器和一个状态位，最大计数值为 32 767。计数器设定值 PV 的数据类型为整数型 INT，寻址范围为：VW，IW，QW，MW，SW，SMW，LW，AIW，T，C，AC，*VD，*AC，*LD 及常数。

计数器用来累计输入脉冲的次数，在实际应用中用来对产品进行计数或完成复杂的逻辑控制任务。计数器的使用和定时器基本相似，编程时各输入端都应有位控制信号，计数器累计它的脉冲输入端信号上升沿的个数。依据设定值及计数器类型决定动作时刻，以便完成计数控制任务。

计数器指令的 LAD 和 STL 格式见表 3-5。

1. 增计数器 CTU（Count Up）

在梯形图中，增计数器以功能框的形式编程，指令名称为 CTU，见表 3-5，它有 3 个输入端：CU、R 和 PV。当复位输入端（R）电路断开，如图 3-8 所示，加计数脉冲输入端（CU）电路由断开变为接通（即 CU 信号的上升沿），计数器计数 1 次，当前值增加 1 个单位，PV 为设定值输入端，当前值达到设定值时，计数器动作，计数器位 ON，当前值可继续计数到 32 767 后停止计数。当复位输入端（R）为 ON 或对计数器执行复位指令，计数器自动复位，计数器位为 OFF，当

表 3-5　计数器的指令格式

格　式	名　　称		
	增计数	增减计数	减计数
LAD	CTU CU　CTU R PV	CTUD CU　CTUD CD R PV	CTD CD　CTD LD PV
STL	CTU C***，PV	CTUD C***，PV	CTD C***，PV

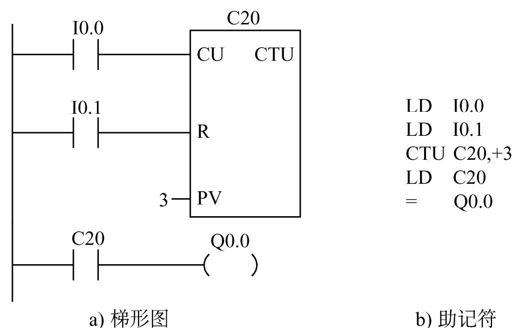


图 3-8　增计数器的梯形图及语句表

前值为零。

2. 增减计数器 CTUD（Count Up/Down）

在梯形图中，增减计数器以功能框的形式编程，指令名称为 CTUD，见表 3-5，CTUD 有 4 个输入端：CU 输入端用于递增计数，CD 输入端用于递减计数，R 输入端用于复位，PV 为设定值输入端。CU 输入的每个上升沿，计数器当前值加 1；CD 输入的每个上升沿，都使计数器当前值减 1，当前值达到设定值时，计数器动作，其状态位为 ON。若复位输入端 R 为 ON，或使用复位指令 R，都可使计数器复位，状态位变为 OFF，并使当前值清 0。

增减计数器当前值计数到 32 767（最大值）后，下一个 CU 输入的上升沿将使当前值跳变为最小值（–32 767）；当前值达到最小值–32 767 后，下一个 CD 输入的上升沿将使当前值跳变为最大值 32 767。如图 3-9 所示为增减计数器的用法。

3. 减计数器 CTD（Count Down）

在梯形图中，减计数器以功能框的形式编程，指令名称为 CTD，见表 3-5，它有 3 个输入端：CD、LD 和 PV。当复位输入端（LD）电路断开（见图 3-10），减计数脉冲输入端（CD）电路由断开变为接通（即 CD 信号的上升沿），计数器计数 1 次，当前值减去 1 个单位，PV 为设定值输入端，当前值减到 0 时，计数器动作，计数器位 ON，计数器的当前值保持为 0。当复位输入端（LD）为 ON 或对

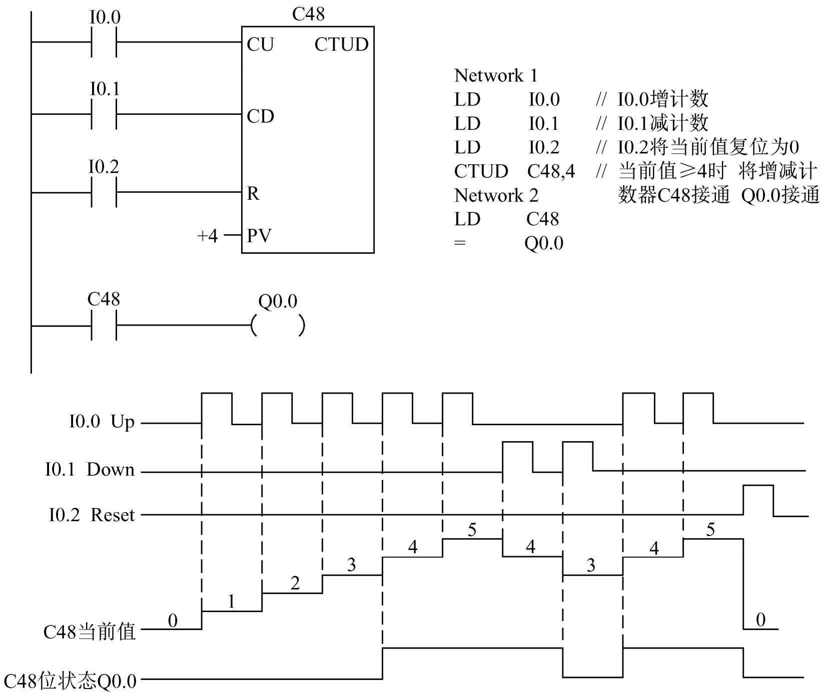


图 3-9 增减计数器的梯形图、语句表及时序图

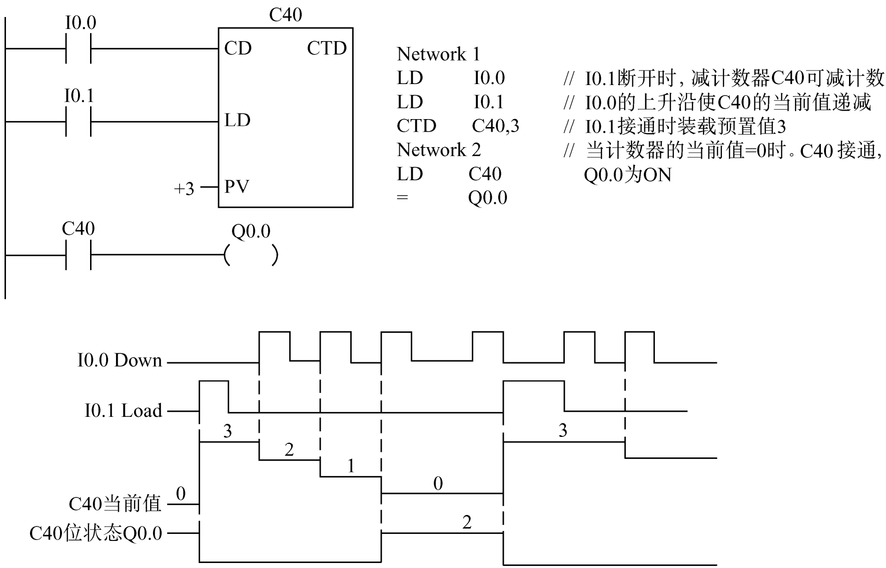


图 3-10 减计数器的梯形图、语句表及时序图

计数器执行复位指令，计数器自动复位，即计数器位为 OFF，当前值为设定值。

4. 计数器计数次数的串级组合

PLC 的单个计数器的计数次数是一定的，或者说是有限的。在 S7-200 PLC 中，单个计数器的最大计数范围是 32 767，当所需计数的次数超过这个最大值时，可通过计数器串级组合的方法来扩大计数器的计数范围。

例如，某产品的生产个数达到 50 万个时，将有一个输出动作，假设 I0.0 为计数开关，I0.1 为清零开关，Q0.0 为 50 万个时的输出位，梯形图程序如图 3-11 所示，50 万个数用一个计数器是实现不了的，这里使用了两个，C1 的设定值是 25000，C2 的设定值是 20，当达到 C2 的设定值时，对 I0.0 的计数次数已达到 25000×20=500000 次。

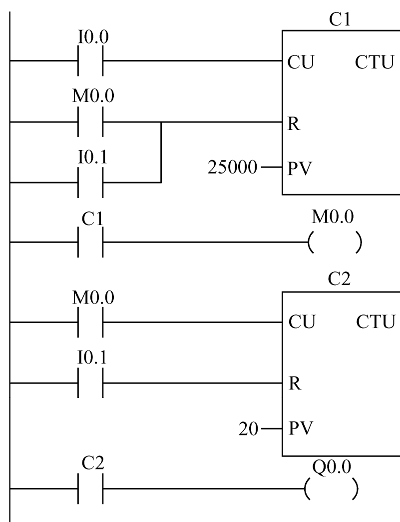


图 3-11　2 个计数器串级组合的梯形图

四、比较指令

比较指令用于两个相同数据类型的有符号数或无符号数 IN1 和 IN2 的比较判断操作。

比较运算符有：等于（=）、大于等于（>=）、小于等于（<=）、大于（>）、小于（<）、不等于（<>），共 6 种比较形式。

在梯形图中，比较指令是以动合触点的形式编程的，在动合触点的中间注明比较参数和比较运算符。触点中间的参数 B、I、D、R 分别表示字节、整数、双字、实数，当比较的结果满足比较关系式给出的条件时，该动合触点闭合。梯形图及语句表中比较指令的基本格式如图 3-12 所示，左为梯形图，右为语句表，图中第一段程序行中有两条比较指令，第一条是计数器 C5 与整数 3 比较，如 C5 中

的计数值与 3 相等，该动合触点将闭合为 ON 状态。指令中的 C5 即是操作数 IN1，3 即是操作数 IN2，触点中间的参数 I 表示与整数比较，运算符是“=”号，说明 IN1 与 IN2 如相等，此触点就为 ON 状态了。后面的第二条是 MB10 与 6 相比较，这条的比较参数是 B，也就是说这是一条字节比较指令，意思是当字节 MB10 中的数据大于等于 6 时条件满足，此触点为 ON 状态，那么当两条指令的条件都满足时线圈 Q0.0 也就为 ON 状态了。第二段程序行中是一条双字比较指令，这里的操作数 IN1 是 0 号高速计数器 HC0，操作数 IN2 是 HC0 的设定值存放地址 SMD42，当两者相等时线圈 Q0.1 为 ON 状态。从这里我们可看出操作数 IN1、操作数 IN2 与比较参数都是统一对应的，不可错用。表 3-6 列出了操作数 IN1 与操作数 IN2 的寻址范围。

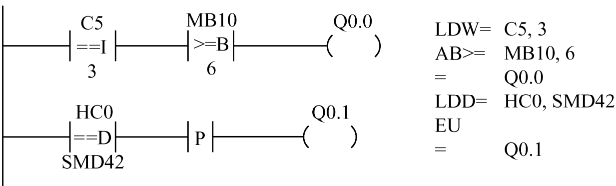


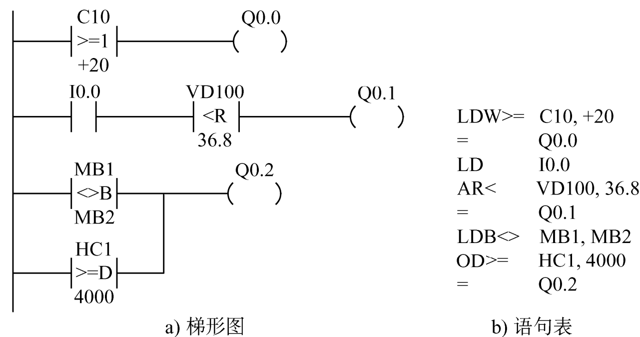
图 3-12 比较指令在梯形图中的基本格式

表 3-6 比较指令的操作数 IN1 和 IN2 的寻址范围

操作数	类型	寻 址 范 围
IN1 IN2	字节	VB，IB，QB，MB，SB，SMB，LB，AC，*VD，*AC，*LD 和常数
	整数	VW，IW，QW，MW，SW，SMW，LW，AIW，T，C，AC，*VD，*AC，*LD 和常数
	双字	VD，ID，QD，MD，SD，SMD，LD，HC，AC，*VD，*AC，*LD 和常数
	实数	VD，ID，QD，MD，SD，SMD，LD，AC，*VD，*AC，*LD 和常数

字节比较指令用于两个无符号的整数字节 IN1 和 IN2 的比较；整数比较指令用于两个有符号的一个字长的整数 IN1 和 IN2 的比较，整数范围为十六进制的 8000 到 7FFF，在 S7-200 PLC 中，用 16#8000～16#7FFF 表示；双字节整数比较指令用于两个有符号的双字长整数 IN1 和 IN2 的比较。双字整数的范围为：16#80000000～16#7FFFFFFF；实数比较指令用于两个有符号的双字长实数 IN1 和 IN2 的比较，正实数的范围为：+1.175495E−38～+3.402823E+38，负实数的范围为：−1.175495E−38～−3.402823E+38。

图 3-13 是一个比较指令使用较多的程序段，从中可以看出：计数器 C10 中的当前值大于等于 20 时，Q0.0 为 ON；VD100 中的实数小于 36.8 且 I0.0 为 ON 时，Q0.1 为 ON，MB1 中的值不等于 MB2 中的值或者高速计数器 HC1 的计数值大于等于 4000 时，Q0.2 为 ON。



a) 梯形图　　b) 语句表

图 3-13　比较指令使用举例

第二节　数据处理指令

一、传送类指令

传送类指令用于在各个编程元件之间进行数据传送。根据每次传送数据的数量，可分为单个传送指令和块传送指令。

1. 单个传送指令 MOVB，MOVW，MOVD，MOVR

单个传送指令每次传送 1 个数据，传送数据的类型分为字节传送、字传送、双字传送和实数传送。表 3-7 列出了单个传送类指令的类别。影响允许输出 ENO 正常工作的出错条件是：SM4.3（运行时间），0006（间接寻址）。IN 和 OUT 的寻址范围见表 3-8。

表 3-7　单个传送类指令表

指令名称	梯形图符号	助记符	指令功能
字节传送 MOV_B	MOV_B EN ENO IN OUT	MOVB IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 个无符号的单字节数据 IN 传送到 OUT 中
字传送 MOV_W	MOV_W EN ENO IN OUT	MOVW IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 个无符号的单字长数据 IN 传送到 OUT 中
双字传送 MOV_DW	MOV_DW EN ENO IN OUT	MOVD IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 个有符号的双字长数据 IN 传送到 OUT 中
实数传送 MOV_R	MOV_R EN ENO IN OUT	MOVR IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 个有符号的双字长实数数据 IN 传送到 OUT 中

表 3-8 传送指令中 IN 和 OUT 的寻址范围

传送	操作数	类型	寻 址 范 围
字节	IN	BYTE	VB，IB，QB，MB，SMB，LB，SB，AC，*AC，*LD，*VD 和常数
	OUT	BYTE	VB，IB，QB，MB，SMB，LB，SB，AC，*AC，*LD，*VD
字	IN	WORD	VW，IW，QW，MW，SMW，LW，SW，AC，*AC，*LD，*VD，T，C 和常数
	OUT	WORD	VW，IW，QW，MW，SMW，LW，SW，AC，*AC，*LD，*VD，T，C
双字	IN	DWORD	VD，ID，QD，MD，SMD，LD，AC，HC，*AC，*LD，*VD 和常数
	OUT	DWORD	VD，ID，QD，MD，SMD，LD，AC，*AC，*LD，*VD
实数	IN	REAL	VD，ID，QD，MD，SMD，LD，AC，HC，*AC，*LD，*VD 和常数
	OUT	REAL	VD，ID，QD，MD，SMD，LD，AC，*AC，*LD，*VD

2. 块传送指令 BMB，BMW，BMD

块传送指令用来进行一次传送多个数据，将最多可达 255 个的数据组成 1 个数据块，数据块的类型可以是字节块、字块和双字块。表 3-9 列出了块传送类指令的类别。影响允许输出 ENO 正常工作的出错条件是：SM4.3（运行时间），0006（间接寻址），0091（数超界）。块传送指令的 IN，N，OUT 的寻址范围见表 3-10。

表 3-9 块传送类指令表

指令名称	梯形图符号	助记符	指令功能
字节块传送 BLKMOV_B	BLKMOV_B EN ENO IN OUT N	BMB IN，OUT，N	当允许输入 EN 有效时，将从输入字节 IN 开始的 *N* 个字节型数据传送到从 OUT 开始的 *N* 个字节存储单元，功能框形式编程
字块传送 BLKMOV_W	BLKMOV_W EN ENO IN OUT N	BMW IN，OUT，N	当允许输入 EN 有效时，将从输入字 IN 开始的 *N* 个字型数据传送到从 OUT 开始的 *N* 个字存储单元，功能框形式编程
双字块传送 BLKMOV_D	BLKMOV_D EN ENO IN OUT N	BMD IN，OUT，N	当允许输入 EN 有效时，将从输入双字 IN 开始的 *N* 个双字型数据传送到从 OUT 开始的 *N* 个双字存储单元，功能框形式编程

表 3-10 块传送指令的 IN，N，OUT 的寻址范围

指令	操作数	类型	寻 址 范 围
BMB	IN、OUT	BYTE	VB，IB，QB，MB，SMB，LB，HC，AC，*AC，*LD，*VD
	N	BYTE	VB，IB，QB，MB，SMB，LB，AC，*AC，*LD，*VD
BMW	IN、OUT	WORD	VW，IW，QW，MW，SMW，LW，AIW，AC，AQW，HC，C，T，*AC，*LD，*VD
	N	BYTE	VB，IB，QB，MB，SMB，LB，AC，*AC，*LD，*VD
BMD	IN、OUT	DWORD	VD，ID，QD，MD，SMD，LD，SD，AC，HC，*AC，*LD，*VD
	N	BYTE	VB，IB，QB，MB，SMB，LB，AC，*AC，*LD，*VD 和常数

二、移位指令

1. *左移和右移指令*

移位指令在 PLC 控制中是比较常用的，根据移位的数据长度可分为字节型移位、字型移位和双字型移位；根据移位的方向可分为左移和右移，还可进行循环移位。指令有右移位指令、左移位指令、循环右移位指令、循环左移位指令。

移位指令的类别见表 3-11，左移或右移指令的特点如下：

表 3-11　移位指令表

指令名称	梯形图符号	助记符	指令功能
字节左移 SHL_B	SHL_B EN ENO IN OUT N	SLB OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将字节型输入数据 IN 左移 *N* 位（*N*≤8）后，送到 OUT 指定的字节存储单元
字节右移 SHR_B	SHR_B EN ENO IN OUT N	SRB OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将字节型输入数据 IN 右移 *N* 位（*N*≤8）后，送到 OUT 指定的字节存储单元
字左移 SHL_W	SHL_W EN ENO IN OUT N	SLW OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将字型输入数据 IN 左移 *N* 位（*N*≤16）后，送到 OUT 指定的字存储单元
字右移 SHR_W	SHR_W EN ENO IN OUT N	SRW OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将字型输入数据 IN 右移 *N* 位（*N*≤16）后，送到 OUT 指定的字存储单元
双字左移 SHL_DW	SHL_DW EN ENO IN OUT N	SLD OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将双字型输入数据 IN 左移 *N* 位（*N*≤32）后，送到 OUT 指定的双字存储单元
双字右移 SHR_DW	SHR_DW EN ENO IN OUT N	SRD OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将双字型输入数据 IN 右移 *N* 位（*N*≤32）后，送到 OUT 指定的双字存储单元

① 被移位的数据是无符号的；

② 在移位时，存放被移位数据的编程元件的移出端与特殊继电器 SM1.1 连接，移出位进入 SM1.1（溢出），另一端自动补 0；

③ 移位次数 *N* 与移位数据的长度有关，如 *N* 小于实际的数据长度，则执行 *N* 次移位。如 N 大于数据长度，则执行移位的次数等于实际数据长度的位数；

④ 移位次数 *N* 为字节型数据。

影响允许输出 ENO 正常工作的出错条件是：SM4.3（运行时间），0006（间接寻址）。

2. *循环左移和循环右移指令*

循环移位的特点如下：

① 被移位时的数据是无符号的；

② 在移位时，存放被移位数据的编程元件的移出端既与另一端连接，又与特殊继电器 SM1.1 连接，移出位在被移到另一端的同时，也进入 SM1.1（溢出）；

③ 移位次数 *N* 与移位数据的长度有关，如 *N* 小于实际的数据长度，则执行 *N* 次移位；如 *N* 大于数据长度，则执行移位的次数为 *N* 除以实际数据长度的余数；

④ 移位次数 *N* 为字节型数据。

如果执行循环移位操作，移出的最后一位的数值存放在溢出位 SM1.1。如果实际移位次数为 0，零标志位 SM1.0 被置为 1。字节操作是无符号的，如果对有符号的字或双字操作，符号位也一起移动。循环移位指令类别见表 3-12。

表 3-12　循环移位指令表

指令名称	梯形图符号	助记符	指令功能
字节循环左移 ROL_B	ROL_B: EN ENO, IN OUT, N	RLB OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将字节型输入数据 IN 循环左移 *N* 位后，送到 OUT 指定的字节存储单元
字节循环右移 ROR_B	ROR_B: EN ENO, IN OUT, N	RRB OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将字节型输入数据 IN 循环右移 *N* 位后，送到 OUT 指定的字节存储单元
字循环左移 ROL_W	ROL_W: EN ENO, IN OUT, N	RLW OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将字型输入数据 IN 循环左移 *N* 位后，送到 OUT 指定的字存储单元
字循环右移 ROR_W	ROR_W: EN ENO, IN OUT, N	RRW OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将字型输入数据 IN 循环右移 *N* 位后，送到 OUT 指定的字存储单元
双字循环左移 ROL_DW	ROL_DW: EN ENO, IN OUT, N	RLD OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将双字型输入数据 IN 循环左移 *N* 位后，送到 OUT 指定的双字存储单元
双字循环右移 ROR_DW	ROR_DW: EN ENO, IN OUT, N	RRD OUT，N	以功能框的形式编程，当允许输入 EN 有效时，将双字型输入数据 IN 循环右移 *N* 位后，送到 OUT 指定的双字存储单元

3. 传送类指令与循环指令应用实例

控制要求：用 1 个按钮控制彩灯循环，方法是第一次按下按钮为起动循环，第二次按下为停止循环，以此为奇数次起动偶数次停止。用另一个按钮控制循环方向，第一次按下左循环，第二次按下右循环，由此交替。假设彩灯初始状态为 00000101，循环移动周期为 1s。

I/O 分配：I0.0—起动、停止按钮　I0.1—左、右循环按钮　Q0.0—Q0.7 彩灯对应位（一个字节）。

程序注释：参见图 3-14，程序中 SM0.1 是个特殊继电器，利用它从 STOP 转为 RUN 只 ON 一个扫描周期的特点为彩灯设置初始值 00000101（16#05），在这用了字节传送指令 MOV_B 将 16#05 送到 QB0 中，按起动按钮 I0.0 为 ON，使 M0.0 置位，时间继电器 T37 开始计时，时间为 1s，到时后是左循环还是右循环要看 M0.1 是否吸合，如吸合为左循环，所用指令为 ROL_B，如没吸合为右循环，所用指令为 ROR_B，而 M0.1 是否吸合由 I0.1 决定，I0.1 单数次 ON 时左循环，I0.1 双数次 ON 时右循环，每隔 1s 循环移动 1 位。

图 3-14　单按钮控制彩灯循环梯形图程序

4. 移位寄存器指令 SHRB

在顺序控制或步进控制中，应用移位寄存器编程是很方便的。

在梯形图中，移位寄存器以功能框的形式编程，指令名称为：SHRB，如图 3-15 所示。它有 3 个数据输入端：DATA 为移位寄存器的数据输入端；S_BIT 为组成移位寄存器的最低位；N 为移位寄存器的长度。

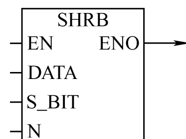


图 3-15 SHRB 梯形图符号

移位寄存器的特点如下：

① 移位寄存器的数据类型无字节型、字型、双字型之分，移位寄存器的长度 N（≤64）由程序指定。

② 移位寄存器的组成：

最低位为：S_BIT；

最高位的计算方法为 MSB={|N|-1+(S_BIT 的位号)}/8；

最高位的字节号：MSB 的商（不包括余数）+S_BIT 的字节号；

最高位的位号：MSB 的余数。

例如：S_BIT=V21.2，N=14，则 MSB=(14-1+2)/8=15/8=1…7

最高位的字节号：21+1=22，最高位的位号：7，最高位为：V22.7

移位寄存器的组成：V21.2～V21.7，V22.0～V22.7，共 14 位。

③ N>0 时，为正向移位，即从最低位向最高位移位。

④ N<0 时，为反向移位，即从最高位向最低位移位。

⑤ 移位寄存器指令的功能是：当允许输入端 EN 有效时，如果 N>O，则在每个 EN 的前沿，将数据输入 DATA 的状态移入移位寄存器的最低位 S_BIT；如果 N<0，则在每个 EN 的前沿，将数据输入 DATA 的状态移入移位寄存器的最高位，移位寄存器的其他位按照 N 指定的方向（正向或反向），依次串行移位。

⑥ 移位寄存器的移出端与 SM1.1（溢出）连接。

移位寄存器指令影响的特殊继电器：SM1.0（零），当移位操作结果为 0 时，SM1.0 自动置位；SM1.1（溢出）的状态由每次移出位的状态决定。

影响允许输出 ENO 正常工作的出错条件为：SM4.3（运行时间），0006（间接寻址），0091（操作数超界），0092（计数区错误）。

在语句表中，移位寄存器的指令格式为：SHRB DATA，S_BIT，N

例：移位寄存器指令的应用如图 3-16 所示。

从图 3-16 中我们可看出，S_BIT=V10.0，N=4>0，最高位为 V10.3。每当我们按下 I0.0 时，I1.0 的状态将从 V10.0 开始移入移位寄存器中，在这里假设移位之前 V10.0 已处于 ON 状态，当第二次按下 I0.0 时，V10.0 的状态已移动到 V10.2，使 V10.2 变为 ON 状态，从而使 Q0.0 也变为 ON 状态。

5. 填充指令 FILL

填充指令 FILL 用于处理字型数据，指令功能是将字型输入数据 IN 填充到从 OUT 开始的 N 个字存储单元，N 为字节型数据。

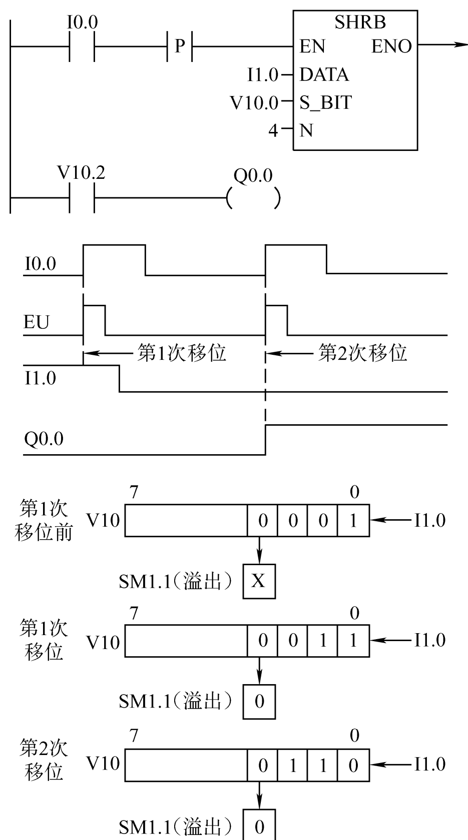


图 3-16　移位寄存器指令应用

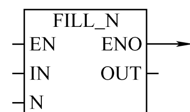


图 3-17　FILL 梯形图符号

在梯形图中，FILL 指令以功能框的形式编程，指令名称为：FILL_N，如图 3-17 所示。当允许输入 EN 有效时，开始填充操作。

例：将 VW100～VW108 这 5 个字都清 0。那除了有效端 EN，0 应放在 IN 端、5 应放在 N 端、VW100 应放在 OUT 端。这样当 EN 端有效时，这 5 个字就都清零了。

影响允许输出 ENO 的出错条件为：SM4.3（运行时间），0006（间接寻址），0091（操作数超界）。

在语句表中，FILL 指令的指令格式为：FILLIN，OUT，N。

第三节　运 算 指 令

随着计算机技术的发展，今天的 PLC 具备了越来越强的运算功能，扩宽了 PLC

的应用领域。运算指令包括算术运算与逻辑运算，算术运算包括加法、减法、乘法、除法及一些常用的数学函数；逻辑运算包括逻辑与、逻辑或、逻辑非、逻辑异或。

一、算术运算指令

在算术运算中，数据类型为整数 INT、双整数 DINT、实数 REAL，对应的运算结果分别为整数、双整数和实数，除法不保留余数。运算结果如超出允许范围，溢出位被置 1。

表 3-13 为常用的加法运算指令，表 3-14 为算术运算指令操作数的寻址范围。

表 3-13　加法运算指令表

指令名称	梯形图符号	助记符	指令功能
整数加法 ADD_I	ADD_I EN　ENO IN1　OUT IN2	+I IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个字型有符号整数 IN1 和 IN2 相加，产生 1 个字型整数和 OUT（字存储单元）。这里 IN2 与 OUT 是同一存储单元
双整数加法 ADD_DI	ADD_DI EN　ENO IN1　OUT IN2	+D IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字型有符号整数 IN1 和 IN2 相加，产生 1 个双字型整数和 OUT（双字存储单元）。这里 IN2 与 OUT 是同一存储单元
实数加法 ADD_R	ADD_R EN　ENO IN1　OUT IN2	+R IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字长实数 IN1 和 IN2 相加，产生 1 个双字长实数和 OUT（双字存储单元）。这里 IN2 与 OUT 是同一存储单元

表 3-14　算术运算指令 IN1、IN2 和 OUT 的寻址范围

指令	操作数	类型	寻 址 范 围
整数	IN1、IN2	INT	VW，IW，QW，MW，SMW，LW，SW，AC，*AC，*LD，*VD，T，C，AIW 和常数
	OUT	INT	VW，IW，QW，MW，SMW，LW，SW，T，C，AC，*AC，*LD，*VD
双整数	IN1、IN2	DINT	VD，ID，QD，MD，SMD，LD，SD，AC，*AC，*LD，*VD，HC 和常数
	OUT	DINT	VD，ID，QD，MD，SMD，LD，SD，AC，*AC，*LD，*VD
实数	IN1、IN2	REAL	VD，ID，QD，MD，SMD，LD，AC，SD，*AC，*LD，*VD 和常数
	OUT	REAL	VD，ID，QD，MD，SMD，LD，AC，*AC，*LD，*VD，SD
完全整数	IN1、IN2	INT	VW，IW，QW，MW，SMW，LW，SW，AC，*AC，*LD，*VD，T，C，AIW 和常数
	OUT	DINT	VD，ID，QD，MD，SMD，LD，SD，AC，*AC，*LD，*VD

加法指令是对两个有符号数进行相加操作，减法指令是对两个有符号数进行

相减操作。表 3-15 为常用的减法运算指令，与加法指令一样，也可分为整数减法指令、双整数减法指令及实数减法指令。运算指令中操作数的寻址范围见表 3-14。

表 3-15 减法运算指令表

指令名称	梯形图符号	助记符	指令功能
整数减法 SUB_I	SUB_I EN ENO IN1 OUT IN2	–I IN2，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个字型有符号整数 IN1 和 IN2 相减，产生 1 个字型整数差 OUT（字存储单元）。这里 IN1 与 OUT 是同一存储单元
双整数减法 SUB_DI	SUB_DI EN ENO IN1 OUT IN2	–D IN2，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字型有符号整数 IN1 和 IN2 相减，产生 1 个双字型整数差 OUT（双字存储单元）。这里 IN1 与 OUT 是同一存储单元
实数减法 SUB_R	SUB_R EN ENO IN1 OUT IN2	–R IN2，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字长实数 IN1 和 IN2 相减，产生 1 个双字长实数差 OUT（双字存储单元）。这里 IN1 与 OUT 是同一存储单元

表 3-16 为常用的乘（除）法运算指令，乘（除）法指令是对两个有符号数进行相乘（除）运算。可分为整数乘（除）法指令、双整数乘（除）法指令、完全整数乘（除）法指令及实数乘（除）法指令。乘法指令中 IN2 与 OUT 为同一个存储单元，而除法指令中 IN1 与 OUT 为同一个存储单元。运算指令中操作数的寻址范围见表 3-14。

表 3-16 乘法、除法运算指令表

指令名称	梯形图符号	助记符	指令功能
整数乘法 MUL_I	MUL_I EN ENO IN1 OUT IN2	×I IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个字型有符号整数 IN1 和 IN2 相乘，产生 1 个字型整数积 OUT（字存储单元）。这里 IN2 与 OUT 是同一存储单元
完全整数乘法 MUL	MUL EN ENO IN1 OUT IN2	MUL IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个字型有符号整数 IN1 和 IN2 相乘，产生 1 个双字型整数积 OUT（双字存储单元）。这里 IN2 与 OUT 的低 16 位是同一存储单元
双整数乘法 MUL_DI	MUL_DI EN ENO IN1 OUT IN2	×D IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字长有符号整数 IN1 和 IN2 相乘，产生 1 个双字型整数积 OUT（双字存储单元）。这里 IN2 与 OUT 是同一存储单元
实数乘法 MUL_R	MUL_R EN ENO IN1 OUT IN2	×R IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字长实数 IN1 和 IN2 相乘，产生 1 个实数积 OUT（双字存储单元）。这里 IN2 与 OUT 是同一存储单元

（续）

指令名称	梯形图符号	助记符	指令功能
整数除法 DIV_I	DIV_I EN ENO IN1 OUT IN2	/ I IN2，OUT	以功能框的形式编程，当允许输入 EN 有效时，用字型有符号整数 IN1 除以 IN2，产生 1 个字型整数商 OUT（字存储单元，不保留余数）。这里 IN1 与 OUT 是同一存储单元
完全整数除法 DIV	DIV EN ENO IN1 OUT IN2	DIV IN2，OUT	以功能框的形式编程，当允许输入 EN 有效时，用字型有符号整数 IN1 除以 IN2，产生 1 个双字型结果 OUT，低 16 位存商，高 16 位存余数。低 16 位运算前存放被除数，这里 IN1 与 OUT 的低 16 位是同一存储单元
双整数除法 DIV_DI	DIV_DI EN ENO IN1 OUT IN2	/ D IN2，OUT	以功能框的形式编程，当允许输入 EN 有效时，将双字长有符号整数 IN1 除以 IN2，产生 1 个整数商 OUT（双字存储单元，不保留余数）。这里 IN1 与 OUT 是同一存储单元
实数除法 DIV_R	DIV_R EN ENO IN1 OUT IN2	/ R IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，用双字长实数 IN1 除以 IN2，产生 1 个实数商 OUT（双字存储单元）。这里 IN1 与 OUT 是同一存储单元

二、增减指令

增减指令又称为自动加 1 或自动减 1 指令。数据长度可以是字节、字、双字，表 3-17 列出了这几种不同数据长度的增减指令。表 3-18 为指令中 IN 及 OUT 的寻址范围。

表 3-17 增减指令表

指令名称	梯形图符号	助记符	指令功能
字节加 1 INC_B	INC_B EN ENO IN OUT	INCB OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 字节长的无符号数 IN 自动加 1，输出结果 OUT 为 1 个字节长的无符号数。指令执行结果：IN+1=OUT
字节减 1 DEC_B	DEC_B EN ENO IN OUT	DECB OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 字节长的无符号数 IN 自动减 1，输出结果 OUT 为 1 个字节长的无符号数。指令执行结果：IN−1=OUT
字加 1 INC_W	INC_W EN ENO IN OUT	INCW OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 字长的有符号数 IN 自动加 1，输出结果 OUT 为 1 个字长的有符号数。指令执行结果：IN+1= OUT
字减 1 DEC_W	DEC_W EN ENO IN OUT	DECW OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 字长的有符号数 IN 自动减 1，输出结果 OUT 为 1 个字长的有符号数。指令执行结果：IN−1= OUT

（续）

指令名称	梯形图符号	助记符	指令功能
双字加 1 INC_D	INC_D EN　ENO IN　OUT	INCD OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 双字长（32 位）的有符号数 IN 自动加 1，输出结果 OUT 为 1 个双字长的有符号数。指令执行结果：IN+1=OUT
双字减 1 DEC_D	DEC_D EN　ENO IN　OUT	DECD OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 双字长（32 位）的有符号数 IN 自动减 1，输出结果 OUT 为 1 个双字长的有符号数。指令执行结果：IN−1=OUT

表 3-18　增减指令中 IN 和 OUT 的寻址范围

指令	操作数	类型	寻 址 范 围
字节增减	IN	BYTE	VB，IB，MB，QB，LB，SB，SMB，*LD，*VD，AC，*AC 和常数
	OUT	BYTE	VB，IB，MB，QB，LB，SB，SMB，*LD，*VD，AC，*AC
字增减	IN	WORD	VW，IW，QW，MW，SMW，LW，SW，AC，*AC，*LD，*VD 和常数
	OUT	WORD	VW，IW，QW，MW，SMW，LW，SW，AC，*AC，*LD，*VD
双字增减	IN	DWORD	VD，ID，QD，MD，SMD，LD，AC，SD，*AC，*LD，*VD 和常数
	OUT	DWORD	VD，ID，QD，MD，SMD，LD，AC，*AC，*LD，*VD，SD

三、逻辑运算指令

逻辑运算指令是对逻辑数（无符号数）进行处理，包括逻辑与、逻辑或、逻辑异或、取反等逻辑操作，数据长度为字节、字、双字。逻辑运算指令见表 3-19。

表 3-19　逻辑运算指令表

指令名称	梯形图符号	助记符	指令功能
字节与 WAND_B	WAND_B EN　ENO IN1　OUT IN2	ANDB IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个 1 字节长的逻辑数 IN1 和 IN2 按位相与，产生 1 字节的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
字节或 WOR_B	WOR_B EN　ENO IN1　OUT IN2	ORB IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个 1 字节长的逻辑数 IN1 和 IN2 按位相或，产生 1 字节的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
字节异或 WXOR_B	WXOR_B EN　ENO IN1　OUT IN2	XORB IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个 1 字节长的逻辑数 IN1 和 IN2 按位异或，产生 1 字节的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
字节取反 INV_B	INV_B EN　ENO IN　OUT	INVB OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 字节长的逻辑数 IN 按位取反，产生 1 字节的运算结果放 OUT。这里 IN 和 OUT 是同一存储单元

（续）

指令名称	梯形图符号	助记符	指令功能
字与 WAND_W	WAND_W EN ENO IN1 OUT IN2	ANDW IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个 1 字长的逻辑数 IN1 和 IN2 按位相与，产生 1 字长的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
字或 WOR_W	WOR_W EN ENO IN1 OUT IN2	ORW IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个 1 字长的逻辑数 IN1 和 IN2 按位相或，产生 1 字长的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
字异或 WXOR_W	WXOR_W EN ENO IN1 OUT IN2	XORW IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个 1 字长的逻辑数 IN1 和 IN2 按位异或，产生 1 字长的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
字取反 INV_W	INV_W EN ENO IN OUT	INVW OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 字长的逻辑数 IN 按位取反，产生 1 字长的运算结果放 OUT。这里 IN 和 OUT 是同一存储单元
双字与 WAND_D	WAND_D EN ENO IN1 OUT IN2	ANDD IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字长的逻辑数 IN1 和 IN2 按位相与，产生 1 个双字长的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
双字或 WOR_D	WOR_D EN ENO IN1 OUT IN2	ORD IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字长的逻辑数 IN1 和 IN2 按位相或，产生 1 个双字长的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
双字异或 WXOR_D	WXOR_D EN ENO IN1 OUT IN2	XORD IN1，OUT	以功能框的形式编程，当允许输入 EN 有效时，将 2 个双字长的逻辑数 IN1 和 IN2 按位异或，产生 1 个双字长的运算结果放 OUT。这里 IN2 和 OUT 是同一存储单元
双字取反 INV_D	INV_D EN ENO IN OUT	INVD OUT	以功能框的形式编程，当允许输入 EN 有效时，将 1 个双字长的逻辑数 IN 按位取反，产生 1 个双字长的运算结果放 OUT。这里 IN 和 OUT 是同一存储单元

表 3-20 为逻辑运算指令中 IN、IN1、IN2 及 OUT 的寻址范围。

表 3-20 逻辑运算指令 IN、IN1、IN2 和 OUT 的寻址范围

指令	操作数	类型	寻 址 范 围
字节逻辑	IN1、IN2 IN	BYTE	VB，IB，MB，QB，LB，SB，SMB，*LD，*VD，AC，*AC 和常数
	OUT	BYTE	VB，IB，MB，QB，LB，SB，SMB，*LD，*VD，AC，*AC

（续）

指令	操作数	类型	寻　址　范　围
字逻辑	IN1、IN2 IN	WORD	VW，IW，QW，MW，SMW，LW，SW，AC，*AC，*LD，*VD，T，C 和常数
	OUT	WORD	VW，IW，QW，MW，SMW，LW，SW，AC，*AC，*LD，*VD，T，C
双字逻辑	IN1、IN2 IN	DWORD	VD，ID，QD，MD，SMD，LD，AC，HC，*AC，*LD，*VD 和常数
	OUT	DWORD	VD，ID，QD，MD，SMD，LD，AC，*AC，*LD，*VD

第四节　转 换 指 令

一、七段显示码指令 SEG

在 S7-200 PLC 中，有一条可直接驱动七段数码管的指令 SEG（Segment），如图 3-18 所示。如果在 PLC 的输出端用 1 字节的前 7 个端口与数码管的 7 个段（a、b、c、d、e、f、g）对应接好，当 SEG 指令的允许输入 EN 有效时，将字节型输入数据 IN 的低 4 位对应的数据（0～F），输出到 OUT 指定的字节单元（只用上前 7 个位），这时 IN 处的数据即可直接通过数码管显示出来。在梯形图中，七段数码指令以功能框的形式编程，在语句表中的指令格式为：SEG IN，OUT。

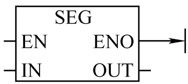


图 3-18　SEG 梯形图符号

二、数据类型转换指令

在进行数据处理时，不同性质的操作指令需要不同数据类型的操作数。数据类型转换指令的功能是将一个固定的数值，根据操作指令对数据类型的需要进行相应类型的转换。表 3-21 列出了几种常用的数据类型转换指令。

表 3-21　数据类型转换指令表

指令名称	梯形图符号	助记符	指令功能
字节到整数 B_I	B_I EN　ENO IN　OUT	BTI IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将字节型输入数据 IN，转换成整数型数据送到 OUT
整数到字节 I_B	I_B EN　ENO IN　OUT	ITB IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将字节型整数输入数据 IN，转换成字节型数据送到 OUT
整数到双整数 I_D	I_D EN　ENO IN　OUT	ITD IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将整数型输入数据 IN，转换成双整数型数据送到 OUT

（续）

指令名称	梯形图符号	助记符	指令功能
双整数到整数 D_I	D_I EN ENO IN OUT	DTI IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将双整数型输入数据 IN，转换成整数型数据送到 OUT
实数到双整数 ROUND	ROUND EN ENO IN OUT	ROUND IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将实数型输入数据 IN，转换成双整数型数据（对 IN 中的小数采取四舍五入），转换结果送到 OUT
实数到双整数 TRUNC	TRUNC EN ENO IN OUT	TRUNC IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将实数型输入数据 IN，转换成双整数型数据（舍去 IN 中的小数部分），转换结果送到 OUT
双整数到实数 DI_R	DI_R EN ENO IN OUT	DTR IN，OUT	以功能框的形式编程，当允许输入 EN 有效时，将双整数型输入数据 IN，转换成实数型数据送到 OUT
整数到 BCD 码 I_BCD	I_BCD EN ENO IN OUT	IBCD OUT	以功能框的形式编程，当允许输入 EN 有效时，将整数型输入数据 IN，转换成 BCD 码输入数据送到 OUT
BCD 码到整数 BCD_I	BCD_I EN ENO IN OUT	BCDI OUT	以功能框的形式编程，当允许输入 EN 有效时，将 BCD 输入数据 IN，转换成整数型输入数据送到 OUT

第五节　程序控制指令

程序控制类指令包括跳转指令、循环指令、顺控继电器指令、子程序指令、结束及暂停指令、看门狗指令，主要用于程序执行流程的控制。对一个扫描周期而言，跳转指令可以使程序出现跨越以实现程序的选择；子程序指令可调用某段子程序，使主程序结构简单清晰，减少扫描时间；循环指令可多次重复执行指定的程序段；顺控继电器指令把程序分成若干个段以实现步进控制；暂停指令可使 CPU 的工作方式发生变化。

以下仅介绍跳转指令、循环指令、子程序指令。

一、跳转指令

跳转指令的功能是根据不同的逻辑条件，有选择地执行不同的程序。利用跳转指令，可以使程序结构更加灵活，减少扫描时间，从而加快了系统的响应速度。

执行跳转指令需要用两条指令配合使用，跳转开始指令 JMPn 和跳转标号指令 LBLn，其中 n 是标号地址，n 的取值范围是 0～255 的字型类型。

使用跳转指令有以下几点注意：

① 由于跳转指令具有选择程序段的功能，在同一程序且位于因跳转而不会被同时执行的两段程序中的同一线圈不被视为双线圈，双线圈指同一程序中，出现对同一线圈的不同逻辑处理现象，这在编程中是不允许的。

② 跳转指令 JMP 和 LBL 必须配合应用在同一个程序块中，即 JMP 和 LBL 可同时出现在主程序中，或者同时出现在子程序中，或者同时出现在中断程序中。不允许从主程序中跳转到子程序或中断程序，也不允许从某个子程序或中断程序中跳转到主程序或其他的子程序或中断程序。

③ 在跳转条件中引入上升沿或下降沿脉冲指令时，跳转只执行一个扫描周期，但若用特殊辅助继电器 SM0.0 作为跳转指令的工作条件，跳转就成为无条件跳转。

在梯形图中，JMPn 以线圈形式编程，LBLn 以功能框形式编程。

例：某食品罐头杀菌工序需一个热水储备罐，如图 3-19 所示，在杀菌处理之前先要给储备罐加水，到达水位后停止加水，开始进蒸汽加热到设定温度关闭进汽阀，当处理信号来到时将热水放入处理罐开始杀菌，杀菌结束后再将热水送回储备罐等待下一次再用。如此循环使用不等的间隔时间，会造成水位与水温的不等，在此就要用跳转指令。

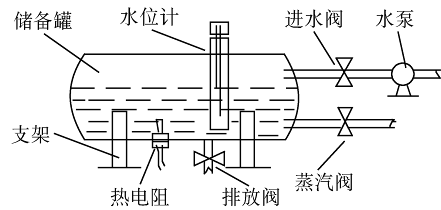


图 3-19　食品罐头杀菌工序热水储备罐示意图

图 3-20 是食品罐头杀菌工序热水储备罐 PLC 控制的对外接线图，当储水开始时，按下起动按钮 SB1，水泵起动（KM1 得电），进水阀（YV1 得电）也同时打

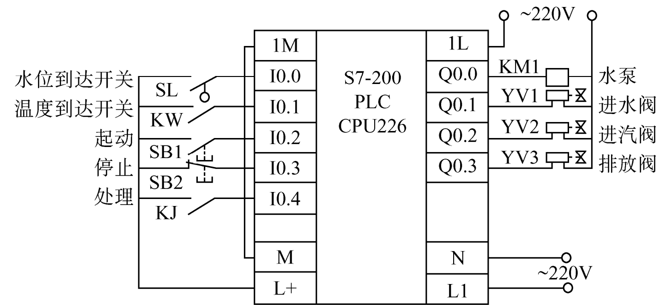


图 3-20　PLC 对外接线图

开。到达设定水位时，水位开关 SL 闭合使水泵停止，进水阀关闭，同时开启进汽阀（YV2）开始加热。到达设定温度时，温度开关 KW 闭合使进汽阀（YV2）关闭。当处理信号来到时，K J 闭合，说明处理罐内已放入罐头可以进行杀菌了，此刻开启排放阀（YV3），将热水放入处理罐。因此储备罐的水是循环再利用的，所以下一次使用时，水位与水温是否还在设定值上是说不准的，这里就需要利用跳转指令进行选择，如图 3-21 所示为此工序的控制程序。

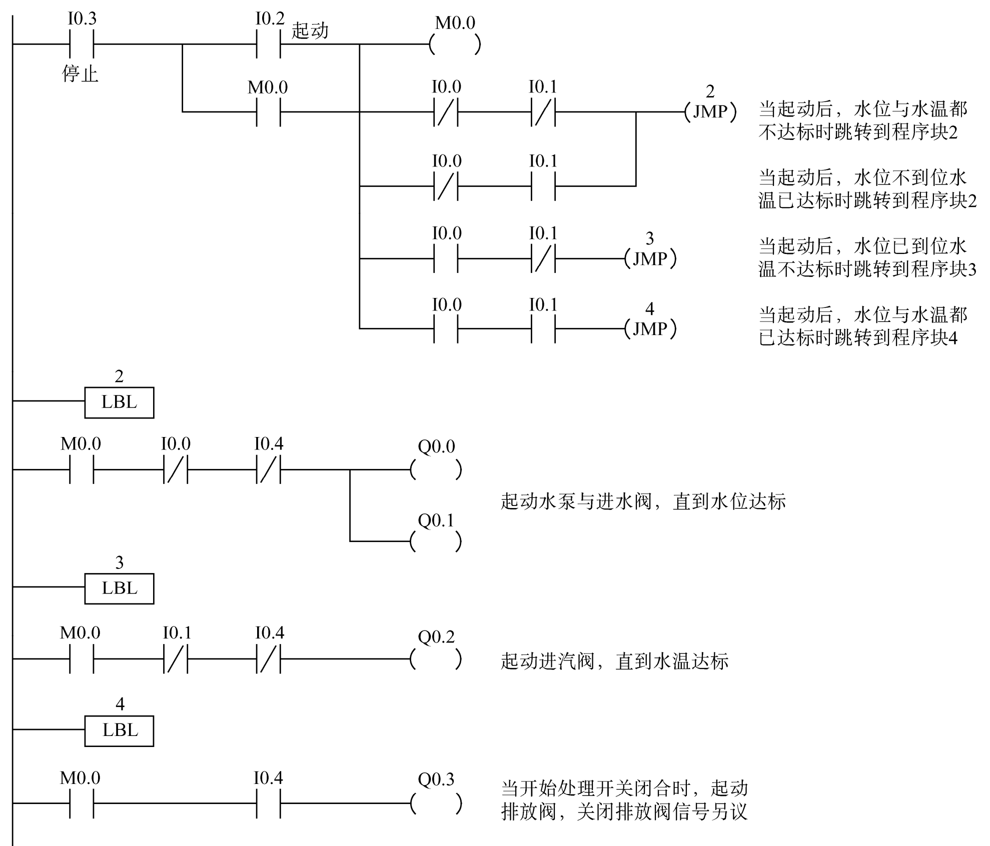


图 3-21　杀菌工序热水储备罐控制程序

二、循环指令

在控制系统中经常遇到对某项任务需重复执行若干次的情况，这时可使用循环指令。循环指令由循环开始指令 FOR 和循环结束指令 NEXT 组成。驱动 FOR 指令的逻辑条件满足时，反复执行 FOR 与 NEXT 之间的程序段。

循环开始指令 FOR 的功能是标记循环体的开始，在梯形图中是以功能框的形

式编程，名称为 FOR，如图 3-22 所示，它有 3 个输入端，分别是 INDX（当前循环计数）、INIT（循环初值）、FINAL（循环终值），它们的数据类型均为整数。循环结束指令 NEXT 的功能是标记循环体的结束，在梯形图中是以线圈的形式编程。

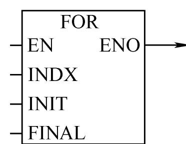


图 3-22　FOR 梯形图符号

FOR 和 NEXT 必须成对使用，在 FOR 和 NEXT 之间构成循环体。当允许输入 EN 有效时，执行循环体，INDX 从 1 开始计数。每执行 1 次循环体，INDX 自动加 1，并且与终值相比较，如果 INDX 大于 FINAL，循环结束。

假设 INIT 是 1，FINAL 是 5，每次执行 FOR 与 NEXT 之间的指令后，INDX 的值加 1，并进行 INDX 与 FINAL 的比较，如果 INDX 大于 5，循环终止，FOR 和 NEXT 之间的指令被执行 5 次。

在语句表中，循环指令的指令格式为：FOR INDX，INIT，FINAL NEXT

三、子程序指令

S7-200 CPU 的控制程序由主程序、子程序和中断程序组成。在 STEP 7Micro/WIN 编程软件的程序编辑器窗口里这三者都有各自独立的页。

在 PLC 的程序设计中，对那些需要经常执行的程序段，设计成子程序的形式，并为每个子程序赋以不同的编号，在程序执行的过程中，可随时调用某个编号的子程序。子程序的调用是有条件的，未调用它时不会执行子程序中的指令，因此使用子程序可以减少扫描时间。使用子程序可以将程序分成容易管理的小块，使程序结构简单清晰，易于查错和维护。

可以在主程序、其他子程序或中断程序中调用子程序，调用某个子程序时将执行该子程序的全部指令，直至子程序结束，然后返回调用它的程序中该子程序调用指令的下一条指令之处。

子程序调用指令 CALL 的功能是将程序执行转移到编号为 n（n=0，1，2，…）的子程序。子程序的入口用指令 SBR n 表示，在子程序执行过程中，如果满足条件返回指令 CRET 的条件，则结束该子程序，返回到主程序原调用处继续执行；否则，将继续执行该子程序到最后一条，也就是无条件返回指令 RET，结束该子程序的运行，返回到主程序。综上所述，进入子程序后，返回时有两种指令，一是有条件返回指令 CRET，一是无条件返回指令 RET。用 STEP 7Micro/WIN 软件编程时，编程人员不用手工输入 RET 指令，当执行子程序到最后一条时，软件会自动将程序返回到主程序原调用处继续执行。

程序控制类指令对合理安排程序的结构、提高程序功能以及实现某些技巧性运算，具有重要的意义。

第六节 特 殊 指 令

一、中断指令

在 S7-200 PLC 中，中断服务程序的调用和处理由中断指令来完成。CPU 提供了中断处理功能，有很多的信息和事件能够引起中断，一般可分为系统内部中断和用户引起的中断。系统的内部中断是由系统来处理的，如编程器、数据处理器及某些智能单元等，都随时会向 CPU 发出中断请求，对于这种中断请求的处理，PLC 是自动完成的，用户不必为此编程。而由用户引起的包括通信中断、高速脉冲串输出中断、外部输入中断、高速计数器中断、定时中断、定时器中断都是需要用户通过设计中断服务程序并设定对应的入口地址来完成的。以上各种中断的先后次序符合优先级排队。

能够用中断功能处理的特定事件称为中断事件。S7-200 PLC 为每个中断事件规定了一个中断事件号。响应中断事件而执行的程序称为中断服务程序，把中断事件号和中断服务程序关联起来才能执行中断处理功能。

中断程序不是由程序调用，而是在中断事件发生时由操作系统调用，这一点是与子程序调用不同的，一旦执行中断程序就会把主程序封存，中断了主程序的正常扫描。中断事件处理完才返回主程序，所以中断程序应尽量短小，否则可能引起主程序控制的设备操作异常。

中断指令主要包括以下几种：

① ENI（全局允许中断）：功能是全局地开放所有被连接的中断事件，允许 CPU 接受所有中断事件的中断请求。在梯形图中，开中断指令以线圈的形式编程，无操作数。

② DISI（全局禁止中断）：功能是全局地关闭所有被连接的中断事件，禁止 CPU 接受所有中断事件的中断请求。在梯形图中，关中断指令以线圈的形式编程，无操作数。

③ ATCH（中断连接）：功能是建立一个中断事件 EVNT 与一个标号为 INT 的中断服务程序的联系，并对该中断事件开放。

中断连接指令在梯形图中以功能框的形式编程，指令名称为 ATCH，如图 3-23 所示。它有两个数据输入端：INT 为中断服务程序的标号，用字节型常数输入；EVNT 为中断事件号，用字节型常数输入。当允许输入有效时，连接与中断事件 EVNT 相关联的 INT 中断程序。

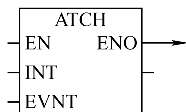


图 3-23 ATCH 梯形图符号

④ DTCH（中断分离）：功能是取消某个中断事件 EVNT 与所有中断程序的关联，并对该中断事件禁止。

中断分离指令在梯形图中以功能框的形式编程，指令名称为 DTCH，如图 3-24 所示，只有一个数据输入端：EVNT，用以指明要被分离的中断事件。当允许输入有效时，切断由 EVNT 指定的中断事件与所有中断程序的联系。

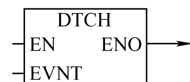


图 3-24　DTCH 梯形图符号

⑤ RETI（中断返回）和 CRETI（中断返回）：功能是，当中断结束时，通过中断返回指令退出中断服务程序，返回到主程序。RETI 是无条件返回指令，CRETI 是有条件返回指令。

例：利用“定时中断”给 8 位彩灯循环左移

控制要求：先设定 8 位彩灯在 QB0 处显示，并设初始值“7”，然后每隔 1s 彩灯循环左移一位。控制按钮选 I0.1 按一次开始，再按一次停止，停止后彩灯全灭。

程序中包括了子程序的调用及中断程序的执行，在子程序中建立了初始化状态并建立与开通了中断事件。应特别注意的是尽管主程序只调用一次子程序，但子程序中的定时中断指令却不停地计时工作，每隔 250ms 产生一次中断，直到按下停止按钮。如图 3-25 为控制程序的梯形图及注释。

S7-200 CPU226 的中断系统中，按中断性质和轻重缓急分配不同的优先级，当多个中断事件同时发出中断请求时，要按表 3-22 所列的优先级顺序进行排队。

表 3-22　中断事件的优先级表

事件号	中断事件描述	组优先级	组内类型	组内优先级
8	通信口 0：单字符接收完成	通信中断最高级	通信口 0	0
9	通信口 0：发送字符完成			0
23	通信口 0：接收信息完成			0
24	通信口 1：接收信息完成		通信口 1	1
25	通信口 1：单字符接收完成			1
26	通信口 1：发送字符完成			1
19	PTO 0 脉冲串输出完成中断	I/O 中断	脉冲串输出	0
20	PTO 1 脉冲串输出完成中断			1
0	I0.0 上升沿中断		外部输入	2
2	I0.1 上升沿中断			3
4	I0.2 上升沿中断			4
6	I0.3 上升沿中断			5

（续）

事件号	中断事件描述	组优先级	组内类型	组内优先级
1	I0.0 下降沿中断	I/O 中断	外部输入	6
3	I0.1 下降沿中断			7
5	I0.2 下降沿中断			8
7	I0.3 下降沿中断			9
12	高速计数器 0：CV=PV（当前值=设定值）			10
27	高速计数器 0：输入方向改变			11
28	高速计数器 0：外部复位			12
13	高速计数器 1：CV=PV（当前值=设定值）			13
14	高速计数器 1：输入方向改变			14
15	高速计数器 1：外部复位			15
16	高速计数器 2：CV=PV		高速计数器	16
17	高速计数器 2：输入方向改变			17
18	高速计数器 2：外部复位			18
32	高速计数器 3：CV=PV（当前值=设定值）			19
29	高速计数器 4：CV=PV（当前值=设定值）			20
30	高速计数器 4：输入方向改变			21
31	高速计数器 4：外部复位			22
33	高速计数器 5：CV=PV（当前值=设定值）			23
10	定时中断 0，SMB34	时基中断 最低级	定时	0
11	定时中断 1，SMB35			1
21	定时器 T32：CT=PT 中断		定时器	2
22	定时器 T96：CT=PT 中断			3

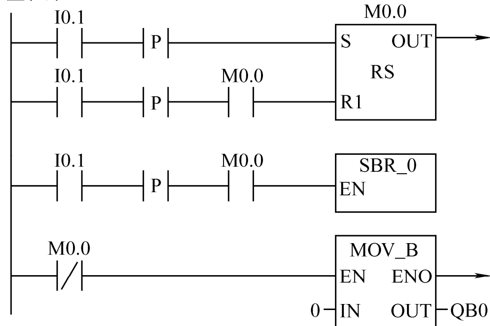
在 S7-200 PLC 的 CPU22X 中，可连接的中断事件及中断事件号见表 3-23。

表 3-23 可连接的中断事件表

CPU 型号	CPU221	CPU222	CPU224	CPU226
可连接的中断事件数	25		31	34
可连接的中断事件号	0～12，19～23，27～33		0～23，27～33	0～33

二、高速计数器指令

普通计数器是按照顺序扫描的方式进行工作，在每个扫描周期中，对计数脉冲只能进行一次累加，计数频率一般仅有几十赫兹。然而，当输入脉冲信号的频

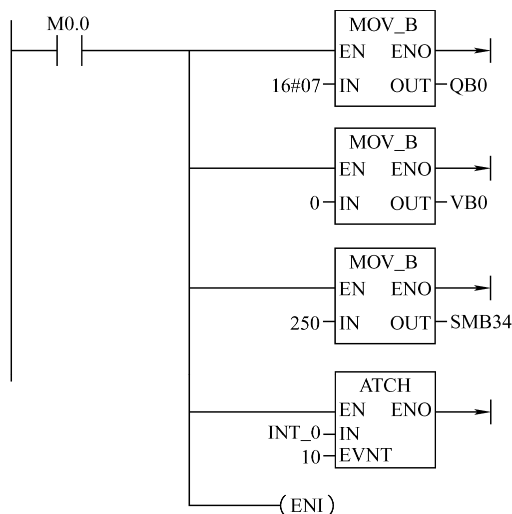


主程序的注释：

第1程序行：用I0.1作为起、停按钮，即按单数次是起动，按双数次是停止。用了RS触发器指令，利用S与R同为1时，R信号状态优先的特点，实现M0.0的ON与OFF的转换。在这里上升沿触发指令的作用至关重要，利用它只给处在自己前面的信号ON一个扫描周期的特点，实现单按钮控制起、停。

第2程序行：这是调用子程序指令，SBR_0指的是调用0号子程序。在I0.1的后面也加了上升沿触发指令，说明这个子程序只需调用一次，对子程序中的程序起到初始化或者说是激活的作用。

第3程序行：程序停止时将QB0清零，也就是说彩灯全灭。



子程序的注释：

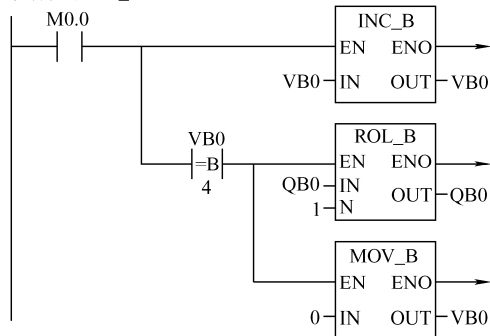
第1条指令：在M0.0闭合的前提下，将16#07（00000111）数据送入QB0字节中用于彩灯显示，准备循环。

第2条指令：将变量存储器VB0整个字节清零，作为计数用。

第3条指令：这是一条能产生定时中断的指令，SMB34是专用于0号中断程序的定时中断，最长时间为255ms，在这里用250是因为本题的彩灯循环间隔是1s，与250ms有整倍数关系，或者还可以是50、100、125。这条指令能达到的目的是计时到250ms，就产生一次中断。

第4条指令：这是一条中断连接指令，它的功能是用0号中断程序执行第10号中断事件。查表可知第10号中断事件即是SMB34产生的定时中断。

第5条指令：允许或者说是开通此中断事件，如没有这条指令将无法进入中断程序。



中断程序的注释：

第1条指令：在M0.0闭合的前提下，当子程序当中的SMB34计时到250ms时，即刻进入中断程序。INC_B指令是字节自动加1指令，这时VB0就会自动加1。

第2条指令：首先是一条字节比较指令，当VB0中的数据为4时，才可执行后面的程序指令，而VB0为4就说明已执行了4次中断程序，次间间隔是250ms，这样4次就是1s了。这时就可以通过左循环指令让QB0左移一位，也就是彩灯左移一位了。

第3条指令：给VB0清零，继续累加计数到下一个1s。

图 3-25　定时中断控制程序的梯形图及注释

率比 PLC 的扫描频率高时，如果仍然采用普通计数器进行累加，必然会丢失很多输入脉冲信号。在 PLC 中，处理比扫描频率高的输入信号的任务是由高速计数器来完成的。

1. 输入端的连接

S7-200 PLC CPU226 拥有 6 个高速计数器 HSC0～HSC5，用以响应快速的脉冲输入信号，可以设置多达 12 种不同的操作模式。用户程序中一旦采用了高速计数器功能，首先要定好高速计数器的号数，也就是在 6 个当中选取，然后就要定模式，因号数与模式相对于 PLC 的输入点都是固定的，见表 3-24。接下来就要编程了，除软件（编程）方面要有相应的初始化设置外，PLC 的输入端也一定要与产生高速脉冲信号的设备，按照已定的号数与模式把导线接好。

在实际工程中，高速计数器大多连接增量型旋转编码器，用于检测位移量和速度等。

旋转编码器一般与被控电动机同轴，每旋转一周可发出一定数量的计数脉冲和一个复位脉冲，作为高速计数器的输入，这种方式的输入信号是不受扫描周期控制的，随来随进，只要用户程序中能利用上送进来的脉冲数就可以了，这就是高速计数器的特点。

每个高速计数器专用的输入点见表 3-24。

表 3-24　高速计数器的输入点

高速计数器编号	输入点
HC0	I0.0，I0.1，I0.2
HC1	I0.6，I0.7，I1.0，I1.1
HC2	I1.2，I1.3，I1.4，I1.5
HC3	I0.1
HC4	I0.3，I0.4，I0.5
HC5	I0.4

表 3-24 中所用到的输入点，如果不使用高速计数器，可作为一般的数字量输入点，有些高速计数器的输入点相互间，或它们与边沿中断（I0.0～I0.3）的输入点有重叠，同一输入点不能同时用于两种不同的功能，但是高速计数器当前模式未使用的输入点可以用于其他功能。例如 HSC0 工作在模式 1 时只使用 I0.0 及 I0.2，那么 I0.1 就可供他用了。在 PLC 的实际应用中，每个输入点的作用是唯一的，不能对某一个输入点分配多个用途，因此要合理分配每一个输入点的用途。

2. 高速计数器的工作模式

工作模式大致分为下面 4 大类：

① 无外部方向输入信号（内部方向控制）的单相加/减计数器（模式 0～2）：可以用高速计数器的控制字节的第 3 位来控制是加还是减。该位是 1 时为加，是 0 时为减；

② 有外部方向输入信号的单相加/减计数器（模式 3～5）：方向输入信号是 1 时为加计数，是 0 时为减计数；

③ 有加计数时钟脉冲和减计数时钟脉冲输入的双相计数器（模式 6～8），也

就是双相增/减计数器，双脉冲输入；

④ A/B 相正交计数器（模式 9～11）：它的两路计数脉冲的相位互差 90º，正转时 A 相在前，反转时 B 相在前。利用这一特点可以实现在正转时加计数，反转时减计数。

3. 高速计数器指令

高速计数器的指令有 2 条：定义高速计数器指令 HDEF 和执行高速计数器指令 HSC。

（1）定义高速计数器指令 HDEF，如图 3-26 所示。

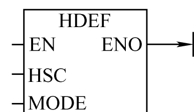


图 3-26　HDEF 梯形图符号

功能是为某个要使用的高速计数器选定一种工作模式。每个高速计数器在使用前，都要用 HDEF 指令来定义工作模式，并且只能定义 1 次。可以用只 ON 一个扫描周期的指令或 SM0.1 调用包含 HDEF 指令的子程序来定义高速计数器，也就是说只激活或者叫初始化一下即可。在梯形图中，HDEF 以功能框的形式编程，它有 2 个数据输入端：HSC 为要使用的高速计数器编号，数据类型为字节型，数据范围为 0～5 的常数，分别对应 HC0～HC5；MODE 为高速计数器的工作模式，数据类型为字节型，数据范围为 0～11 的常数，分别对应 12 种工作模式。当允许输入 EN 有效时，为指定的高速计数器 HSC 定义工作模式 MODE。

（2）执行高速计数器指令 HSC，如图 3-27 所示。

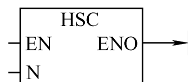


图 3-27　HSC 梯形图符号

功能是根据与高速计数器相关的特殊继电器确定的控制方式和工作状态，使高速计数器的设置生效，按照指定的工作模式执行计数操作。

在梯形图中，HSC 以功能框的形式编程，它有一个数据输入端 N：N 为高速计数器的编号，数据类型为字型，数据范围为 0～5 的常数，分别对应 HC0～HC5。当允许输入 EN 有效时，起动 N 所对应的 HC0～HC5 之一。

4. 高速计数器的控制字节

在使用高速计数器时，用 HDEF 指令定工作模式，用 HSC 指令定开启哪个高速计数器，然后还要对高速计数器的动态参数进行编程。各高速计数器均有一个特殊继电器的控制字节 SMB，通过对控制字节指定位的编程，确定高速计数器的工作方式，各位的意义见表 3-25。执行 HSC 指令时，CPU 检查控制字节及有关的当前值与设定值。执行 HDEF 指令之前必须将控制位设置成需要的状态，否则高速计数器将选用模式的默认设置。一旦执行了 HDEF 指令，设置的控制位就不能再改变，除非 CPU 进入停止模式。

5. 高速计数器的数值寻址

每个高速计数器都有一个初始值和一个设定值，它们都是 32 位有符号整数。初始值是高速计数器计数的起始值；设定值是高速计数器运行的目标值，当实际

计数值等于设定值（见表 3-22）时会发生一个内部中断事件。必须先设置控制字节（见表 3-25）以允许装入新的初始值和设定值，并且把初始值和设定值存入特殊存储器中，然后执行 HSC 指令使新的初始值和设定值有效。高速计数器各种数值存放处见表 3-26。当前值也是一个 32 位的有符号整数，例如，表中的 HSC0 的当前值，在程序里可从 HC0 中直接读出。

表 3-25 高速计数器的控制字节

HC0	HC1	HC2	HC3	HC4	HC5	描 述
SM37.0	SM47.0	SM57.0	—	SM147.0	—	0=复位信号高电平有效，1=低电平有效
—	SM47.1	SM57.1	—	—	—	0=起动信号高电平有效，1=低电平有效
SM37.2	SM47.2	SM57.2	—	SM147.2	—	0=4 倍频模式，1=1 倍频模式
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	0=减计数方向，1=增计数方向
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	0=不改变计数方向，1=可改变计数方向
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	0=不改变设定值，1=可改变设定值
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	0=不改变当前值，1=可改变当前值
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	0=禁止高速计数器，1=允许高速计数器

表 3-26 高速计数器的数值寻址

计数器号	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
初始值	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
设定值	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162
当前值	HC0	HC1	HC2	HC3	HC4	HC5

三、通信指令

PLC 的通信包括 PLC 之间、PLC 与上位计算机之间以及 PLC 与其他智能设备之间的通信。PLC 与计算机可以直接或通过通信处理单元、通信转换器相连构成网络，以实现信息的交换。

1. S7-200 PLC 的网络通信协议

在进行网络通信时，通信双方必须遵守约定的规程，这些为交换信息而建立的规程称为通信协议。

S7-200 系列的 PLC 主要用于现场控制，在主站和从站之间的通信可以采用 3 个标准化协议和 1 个自由口协议。

① PPI（Point to Point Interface）协议，也就是点对点接口协议。

② MPI（Multi Point Interface）协议，也就是多点接口协议。

③ PROFIBUS 协议，用于分布式 I/O 设备的高速通信。

④ 用户定义的协议，也就是自由口协议。

其中的 PPI 协议是 SIEMENS 公司专为 S7-200 系列 PLC 开发的通信协议，是主/从协议，利用 PC/PPI 电缆，将 S7-200 系列的 PLC 与装有 STEP-7 Micro/WIN32 编程软件的计算机连接起来，组成 PC/PPI（单主站）的主/从网络连接。

本节中只介绍 PPI 协议。

网络中的 S7-200 PLC CPU 均为从站，其他 CPU、编程器或人机界面 HMI（如 TD200 文本显示器）为主站。

如果在用户程序中指定某个 S7-200 PLC CPU 为 PPI 主站模式，则在 RUN 工作方式下，可以作为主站，它可以用相关的通信指令读写其他 PLC 中的数据；与此同时，它还可以作为从站响应来自于其他主站的通信请求。

对于任何一个从站，PPI 不限制与其通信的主站的数量，但是在网络中最多只能有 32 个主站。

2. 通信设备

（1）通信端口　S7-200 系列 PLC 中的 CPU226 型机有 2 个 RS-485 端口，外形为 9 针 D 型，分别定义为端口 0 和端口 1，作为 CPU 的通信端口，通过专用电缆可与计算机或其他智能设备及 PLC 进行数据交换。

（2）网络连接器　网络连接器用于将多个设备连接到网络中。一种是连接器的两端只是个封闭的 D 型插头，可用来两台设备间的一对一通信；另一种是在连接器两端的插头上还设有敞开的插孔，可用来连接第三者，实现多设备通信。

（3）PC/PPI 电缆　用此电缆连接 PLC 主机与计算机及其他通信设备，PLC 主机侧是 RS-485 接口，计算机侧是 RS-232 接口。当数据从 RS-232 传送到 RS-485 时，PC/PPI 电缆是发送模式，反之是接收模式。

3. 通信指令

（1）PPI 主站模式设定　在 S7-200 PLC 的特殊继电器 SM 中，SMB30（SMB130）是用于设定通信端口 0（通信端口 1）的通信方式。由 SMB30（SMB130）的低 2 位决定通信端口 0（通信端口 1）的通信协议。只要将 SMB30（SMB130）的低 2 位设置为 2#10，就允许该 PLC 主机为 PPI 主站模式，可以执行网络读写指令。

（2）PPI 主站模式的通信指令　S7-200 PLC CPU 提供网络读写指令，用于 S7-200 PLC CPU 之间的联网通信。网络读写指令只能由在网络中充当主站的 CPU 执行，或者说只给主站编写读写指令，就可与其他从站通信了；从站 CPU 不必做通信编程，只需准备通信数据，让主站读写（取送）有效即可。

在 S7-200 的 PPI 主站模式下，网络通信指令有两条：NETR 和 NETW。

① 网络读指令 NETR（Net Read），如图 3-28 所示。网络读指令通过指定的通信口（主站上 0 口或 1 口）从

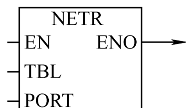


图 3-28　NETR 梯形图符号

其他 CPU 中指定地址的数据区读取最多 16 字节的信息，存入本 CPU 中指定地址的数据区。

在梯形图中，网络读指令以功能框形式编程，指令的名称为：NETR。当允许输入 EN 有效时，初始化通信操作，通过指定的端口 PORT，从远程设备接收数据，将数据表 TBL 所指定的远程设备区域中的数据读到本 CPU 中。TBL 和 PORT 均为字节型，PORT 为常数。

PORT 处的常数只能是 0 或 1，如是 0，就要将 SMB30 的低 2 位设置为 2#10；如是 1，就要将 SMB130 的低 2 位设置为 2#10，这里要与通信端口的设置保持一致。

TBL 处的字节是数据表的起始字节，可以由用户自己设定，但起始字节定好后，后面的字节就要接连使用，形成列表，每字节都有自己的任务，见表 3-27。NETR 指令最多可以从远程设备上接收 16B 的信息。

表 3-27　数据表（TBL）格式

字节偏移地址	字 节 名 称	描　　述
0	状态字节	反映网络通信指令的执行状态及错误码
1	远程设备地址	被访问的 PLC 从站地址
2	远程设备的数据指针	被访问数据的间接指针 指针可以指向 I，Q，M 和 V 数据区
3		
4		
5		
6	数据长度	远程设备被访问的数据长度
7	数据字节 0 数据字节 1 ⋮ 数据字节 15	执行 NETR 指令后，存放从远程设备接收的数据 执行 NETW 指令前，存放要向远程设备发送的数据
8		
…		
22		

在语句表中，NETR 指令的指令格式：NETR TBL，PORT

② 网络写指令 NETW（Net Write），如图 3-29 所示。网络写指令通过指定的通讯口（主站上 0 口或 1 口），把本 CPU 中指定地址的数据区内容写到其他 CPU 中指定地址的数据区内，最多可以写 16B 的信息。

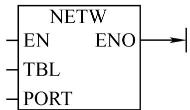


图 3-29　NETW 梯形图符号

在梯形图中，网络写指令以功能框形式编程，指令的名称为：NETW。当允许输入 EN 有效时，初始化通信操作，通过指定的端口 PORT，将数据表 TBL 所指定的本 CPU 区域中的数据发送到远程设备中。TBL 和 PORT 均为字节型，PORT 为常数。数据表 TBL 见表 3-27。

NETW 指令最多可以从远程设备上接收 16B 的信息。

在语句表中，NETW 指令的指令格式：NETW TBL，PORT

在一个应用程序中，使用 NETR 和 NETW 指令的数量不受限制，但是不能同时激活 8 条以上的网络读写指令（例如：同时激活 6 条 NETR 和 3 条 NETW 指令）。

数据表 TBL 共有 23 字节，表头（第一字节）是状态字节，它反映网络通信指令的执行状态及错误码，各个位的意义如下：

MSB　　　　　　　　　　　　　　　　LBS

D	A	E	O	E1	E2	E3	E4

D 位：操作完成位。　　0：未完成，　1：已经完成

A 位：操作排队有效位。　0：无效，　1：有效

E 位：错误标志位。　　0：无错误，　1：有错误

E1，E2，E3，E4 为错误编码。如果执行指令后，E 位为 1，则由 E1E2E3E4 反应一个错误码，编码及说明见表 3-28。

表 3-28　错误编码表

E1E2E3E4	错误码	说　明
0000	0	无错误
0001	1	时间溢出错误：远程设备不响应
0010	2	接收错误：奇偶校验错，响应时帧或检查时出错
0011	3	离线错误：相同的站地址或无效的硬件引发冲突
0100	4	队列溢出错误：同时激活了 8 个以上的网络通信指令
0101	5	违反通信协议：没有在 SMB30 中设置允许 PPI 协议而使用网络指令
0110	6	非法参数：NETR 或 NETW 中包含有非法或无效的值
0111	7	没有资源：远程设备忙，如正在上装或下装程序
1000	8	第 7 层错误：违反应用协议
1001	9	信息错误：错误信息的数据地址或不正确的数据长度

四、PID 回路控制指令

在过程控制中，经常涉及模拟量的控制，比如温度、压力和流量控制等。为了使控制系统稳定准确，要对模拟量进行采样检测，形成闭环控制系统。检测的对象是被控物理量的实际数值，也称为过程变量；用户设定的调节目标值，也称为给定值。控制系统对过程变量与给定值的差值进行 PID 运算，根据运算结果，形成对模拟量的控制作用。

PID 即比例/积分/微分，在闭环系统中，PID 调节器的控制作用是使系统在稳定的前提下，偏差量最小，并自动消除各种因素对控制效果的扰动。

1. PID 回路表

在 S7-200 PLC 中，通过 PID 回路指令来处理模拟量是非常方便的，PID 功能的核心是 PID 指令。PID 指令需要为其指定一个以 V 变量存储区地址开始的 PID 回路表、PID 回路号。PID 回路表提供了给定和反馈，以及 PID 参数等数据入口，PID 运算的结果也在回路表输出，见表 3-29。

表 3-29 PID 指令回路表

偏移地址	参数名	数据格式	类 型	描 述
0	PV_n	双字，实数	输入	过程变量当前值，应在 0.0～1.0 之间
4	SP_n		输入	给定值，应在 0.0～1.0 之间
8	M_n		输入/输出	输出值，应在 0.0～1.0 之间
12	K_c		输入	比例增益，常数，可正可负
16	T_s		输入	采样时间，单位为 s，应为正数
20	T_I		输入	积分时间常数，单位为 min，应为正数
24	T_D		输入	微分时间常数，单位为 min，应为正数
28	MX		输入/输出	积分项前值，应在 0.0～1.0 之间
32	PV_{n-1}		输入/输出	最近一次 PID 运算的过程变量值

PID 回路有两个输入量，即给定值（SP）与过程变量（PV）。给定值通常是固定的值，过程变量是经 A/D 转换和计算后得到的被控量的实测值。给定值与过程变量都是现实存在的值，对于不同的系统，它们的大小、范围与工程单位有很大的区别。在回路表中它们只能被 PID 指令读取，而不能改写。PID 指令对这些量进行运算之前，还要进行标准化转换。每次完成 PID 运算后，都要更新回路表内的输出值 M_n，它被限制在 0.0～1.0 之间。从手动控制切换到 PID 自动控制方式时，回路表中的输出值可以用来初始化输出值。

增益 K_c 为正时为正作用回路，反之为反作用回路。如果不想要比例作用，应将回路增益 K_c 设为 0.0，对于增益为 0.0 的积分或微分控制，如果积分或微分时间为正，为正作用回路，反之为反作用回路。

如果使用积分控制，上一次的积分值 MX（积分和）要根据 PID 运算的结果来更新，更新后的数值作为下一次运算的输入。MX 也应限制在 0.0～1.0 之间，每次 PID 运算结束时，将 MX 写入回路表，供下一次 PID 运算使用。

2. PID 参数的整定方法

为执行 PID 指令，要对某些参数进行初始化设置，也可称为整定，参数整定对控制效果的影响非常大，PID 控制器有 4 个主要的参数 T_s、K_c、T_I 和 T_D 需要整定。

在 P、I、D 这三种控制作用中，比例（P）部分与误差在时间上是一致的，只要误差一出现，比例部分就能及时地产生与误差成正比的调节作用，具有调节及

时的特点。比例系数 K_c 越大，比例调节作用越强，但过大会使系统的输出量振荡加剧，稳定性降低。

积分（I）部分与误差的大小和误差的历史情况都有关系，只要误差不为零，控制器的输出就会因积分作用而不断变化，一直要到误差消失，系统处于稳定状态时，积分部分才不再变化，因此积分部分可以消除稳态误差，提高控制精度。但是积分作用的动作缓慢，滞后性强，可能给系统的动态稳定性带来不良影响。积分时间常数 T_I 增大时，积分作用减弱，系统的动态稳定性可能有所改善，但是消除稳态误差的速度减慢。

微分（D）部分反映了被控量变化的趋势，微分部分根据它提前给出较大的调节作用。它较比例调节更为及时，所以微分部分具有超前和预测的特点。微分时间常数 T_D 增大时，可能会使超调量减小，动态性能得到改善，但是抑制高频干扰的能力下降。如果 T_D 过大，系统输出量可能出现频率较高的振荡。

为使采样值能及时反映模拟量的变化，T_s 越小越好。但是 T_s 太小会增加 CPU 的运算工作量，相邻两次采样的差值几乎没有什么变化，所以也不宜将 T_s 取得过小。表 3-30 给出过程控制中采样周期的经验数据。

表 3-30　采样周期的经验数据

被控制量	流量	压力	温度	液位
采样周期/s	1～5	3～10	15～20	6～8

3. PID 回路控制指令

S7-200 PLC 的 PID 指令没有设置控制方式，执行 PID 指令时为自动方式；不执行 PID 指令时为手动方式。PID 指令的功能是进行 PID 运算。

当 PID 指令的允许输入 EN 有效时，即进行手动/自动控制切换，开始执行 PID 指令。为了保证在切换过程中无扰动、无冲击，在转换前必须把当前的手动控制输出值写入回路表的参数 M_n。并对回路表内的值进行下列操作：

① 使 SP_n（给定值）$=PV_n$（过程变量）。

② 使 PV_{n-1}（前一次过程变量）$= PV_n$（过程变量的当前值）。

③ 使 MX（积分和）$=M_n$（输出值）。

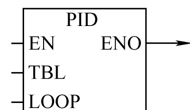


图 3-30　PID 梯形图符号

在梯形图中，PID 指令以功能框的形式编程，指令名称为 PID，如图 3-30 所示。在功能框中有两个数据输入端：TBL 是回路表的起始地址，是由变量寄存器 VB 指定的字节型数据；LOOP 是回路的编号，是 0～7 的常数。当允许输入 EN 有效时，根据 PID 回路表中的输入信息和组态信息，进行 PID 运算。在一个应用程序中，最多可以使用 8 个 PID 控制回路，一个 PID 控制回路只能使用 1 条 PID 指令，不同的 PID 指令不能使用相同的回路编号。

第四章　应 用 实 例

在理解了 PLC 的工作原理，熟悉了 PLC 的结构组成及指令系统之后，就要编写程序了。因为我们学习 PLC 的最终目的是为生产服务，为工程服务，为自动控制系统服务。通过实例编程巩固基础知识，检验掌握程度，构筑应用环境，强化工程意识，进一步提高编程水平和应用能力。

本章选题均是以 S7-200 PLC 系列 CPU226 为样机，用 STEP 7 软件进行编程，程序仅供参考。重要的是明确控制要求，了解指令的使用方法，实现控制目的。

例一、运料小车延时正、反转控制

1. 控制要求

运料小车由三相交流异步电动机驱动。当按下正转起动按钮 SB1 时，如果小车处于停止状态，则立即正转运行，直至碰到正向限位开关 SQ1 后停止；如果小车处于反转运行状态，则先使反向停止，10s 后小车正转运行，直至碰到正向限位开关 SQ1 后停止。当按下反转起动按钮 SB2 时，如果小车处于停止状态，则立即反转运行，直至碰到反向限位开关 SQ2 后停止。如果小车处于正转运行状态，则先使正向停止，10s 后小车反转运行，直至碰到反向限位开关 SQ2 后停止。任何时候按下停止按钮 SB3，小车停止运行。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，进一步定出 PLC 的控制接线图如图 4-1 所示。小车运行示意图如图 4-2 所示。

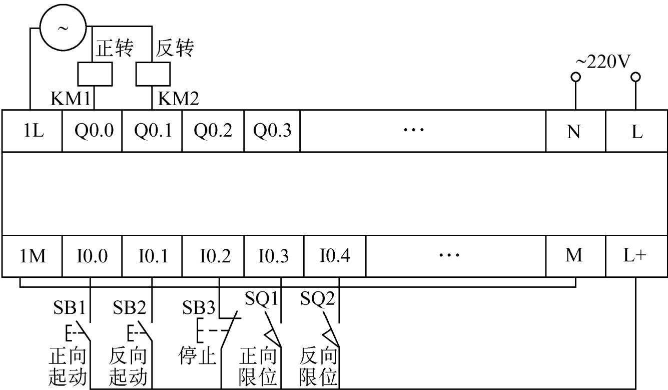


图 4-1　运料小车的 PLC 控制接线图

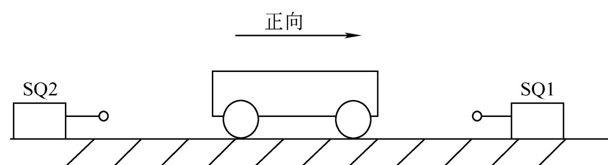


图 4-2　小车运行示意图

（2）控制程序梯形图如图 4-3 所示。

（3）程序的语句表及注释如下：

```
Network 1               // 程序注释
LD      I0.2
LPS
LD      I0.0
AN      M0.0
LD      Q0.0
AN      M0.2
OLD
O       M0.1
ALD
AN      M0.2
AN      Q0.1
AN      I0.3
=       Q0.0            // 正向控制，如反向正在运行需等 10s 后正向才能起动
LRD
LD      I0.0
A       Q0.1
O       M0.0
ALD
AN      T37
=       M0.0            // 按正向按钮，如已有反向先断开反向
LRD
LPS
A       M0.0
TON     T37, 100        // 正向按钮按后经延时再起动
LRD
```

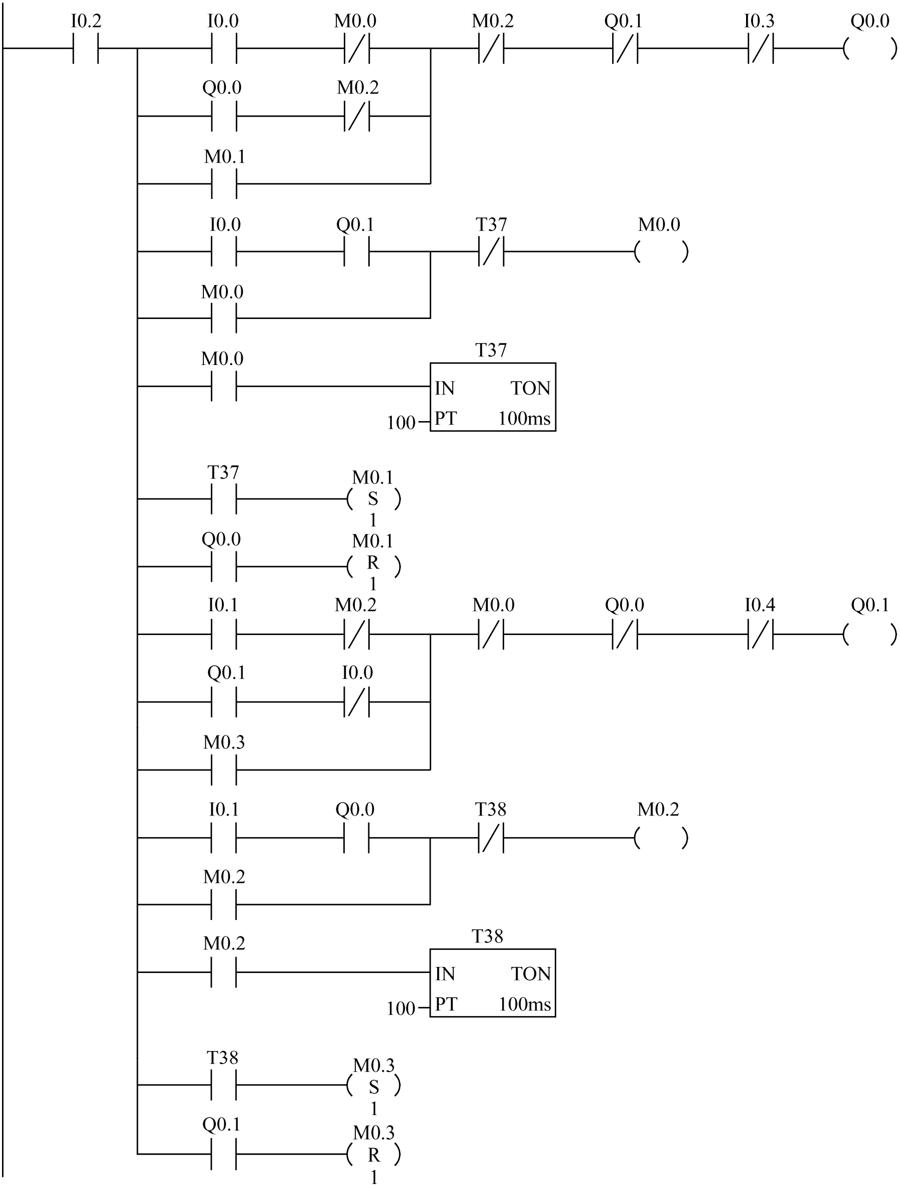


图 4-3 小车运行控制程序梯形图

```
A      T37
S      M0.1, 1
LPP
A      Q0.0
```

```
R      M0.1, 1
LRD
LD     I0.1
AN     M0.2
LD     Q0.1
AN     I0.0
OLD
O      M0.3
ALD
AN     M0.0
AN     Q0.0
AN     I0.4
=      Q0.1      // 反向控制，如正向正在运行需等 10s 后反向才能起动
LRD
LD     I0.1
A      Q0.0
O      M0.2
ALD
AN     T38
=      M0.2      // 按反向按钮，如正在正向运转先断开正向
LPP
LPS
A      M0.2
TON    T38, 100  // 反向按钮按后经延时再起动
LRD
A      T38
S      M0.3, 1
LPP
A      Q0.1
R      M0.3, 1
```

例二、电动机星形-三角形减压起动控制

1. 控制要求

如图 4-4 所示为单台电动机Y-△减压起动控制线路图，将其用 PLC 改造，要求画出 PLC 对外 I/O 接线图，并编写出 PLC 梯形图程序。

2. 程序设计

电动机Y-△减压起动是大家最熟悉的一种减压起动方式了，体现为方法简单，安装维护方便，经济实惠。控制线路也有很多种，不管是哪种控制结果是一样的，其本质就是三个接触器先是第一与第二得电动作，形成电动机的星起动，隔 3s 再是第一与第三得电动作，形成电动机的三角形运行。

用 PLC 程序来实现这一控制过程，关键在于最好不要在一个周期内形成三个接触器都能得电动作的程序。

图 4-4 电动机Y-△减压起动控制线路图

（1）根据控制要求，PLC 的 I/O 接线图如图 4-5 所示。

图 4-5 PLC 的 I/O 接线图

（2）第一方案控制程序梯形图如图 4-6 所示。

图 4-6 电动机Y-△减压起动控制程序梯形图（一）

（3）第二方案控制程序梯形图如图 4-7 所示。

方案二：

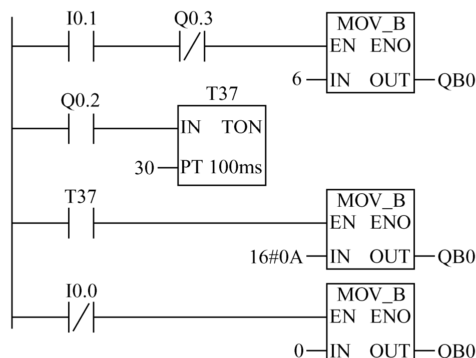


图 4-7　电动机Y-△减压起动控制程序梯形图（二）

3. 编程体会

由方案一得知，（1）为了使所编的程序都放在一个网络里，便于查找监视，程序的整体结构就会出现头轻脚重，按要求是最不允许的一种结构形式，如图 4-6 所示。在一个网络中，将停止按钮放在最前面刚好能形成只有一个程序行与总母线相连的要求，这样给编者带来方便，但给 PLC 操作系统带来了负担，增加了扫描时间，所以如程序较长就尽量不要这样做。（2）利用 PLC 周期性逐行扫描的特点，使程序既简捷又能实现控制要求。

由方案二得知，传送指令的输入信号使能端（EN）所要求的信号类型是脉冲型的，即使所加信号是连续型的，如后面又有传送指令形成，仍然对其进行覆盖。

例三、带式运输机可重复顺序起动、逆序停止控制

1. 控制要求

设计一种控制系统，能够实现多级带式运输机的可重复延时顺序起动、停止控制，所有带式运输机均由三相交流异步电动机驱动。当按下起动按钮 SB1 时，1#带式运输机立即起动运行，延时 5s 后，2#带式运输机起动运行，延时 10s 后，3#带式运输机起动运行，延时 15s 后，4#带式运输机起动运行。任何时候按下停止按钮 SB2，带式运输机逆起动顺序停止，相隔延时均为 8s，直至所有带式运输机均停止运行。在电动机逆序停止过程中，如果按下起动按钮 SB1，则停止过程立即中断，带式运输机按照上述起动规则又可顺序延时起动，延时时间从起动按钮按下时刻算起。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，运输机可重复顺序起动、逆序停止 PLC 控制接线图如图 4-8 所示。

（2）第一方案控制程序梯形图如图 4-9 所示。

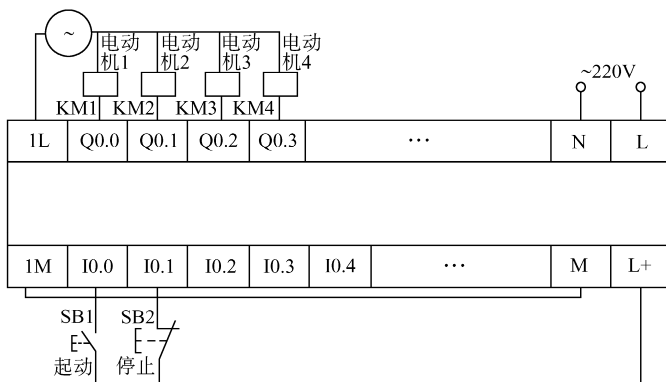


图 4-8 可重复顺序起动、逆序停止 PLC 控制接线图

方案一：

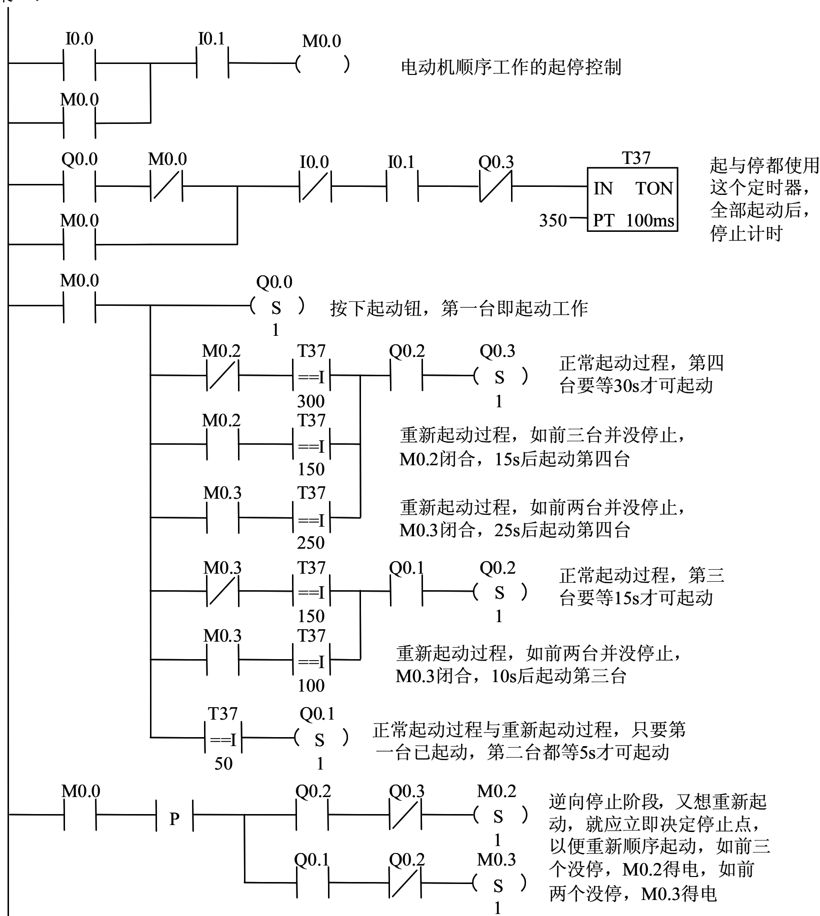


图 4-9 可重复顺起、逆停控制程序梯形图（一）

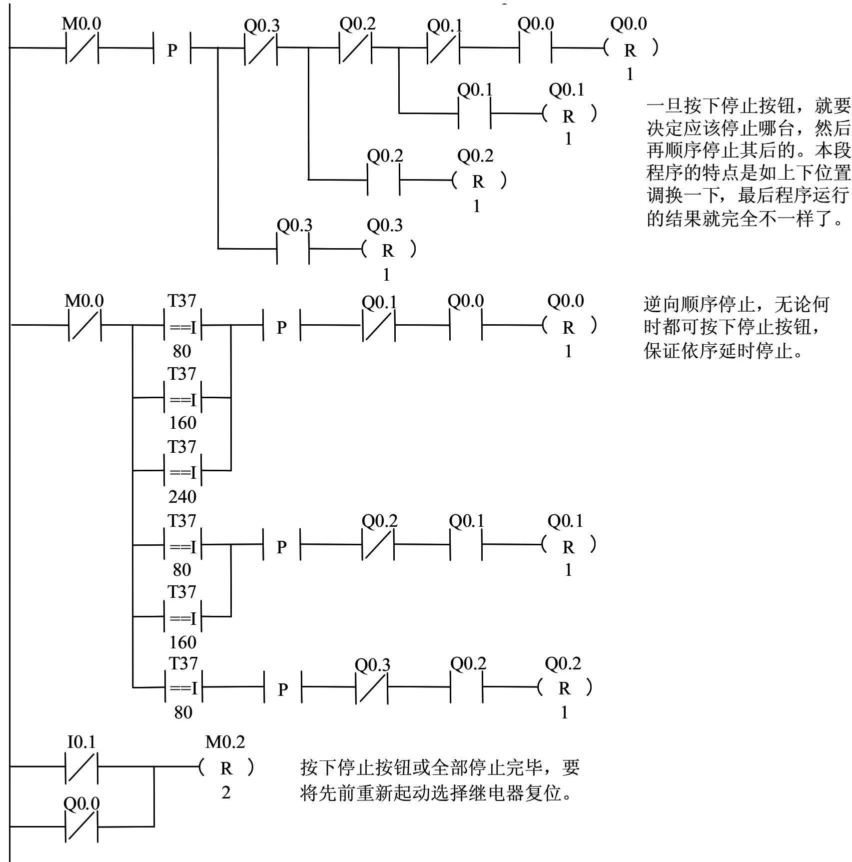


图 4-9　可重复顺起、逆停控制程序梯形图（一）（续）

（3）第二方案控制程序梯形图如图 4-10 所示。

方案二：

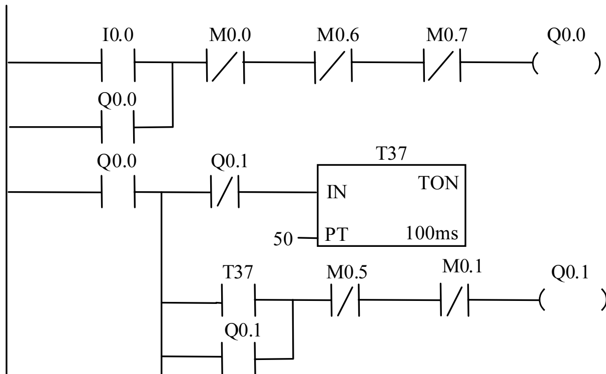


图 4-10　可重复顺起、逆停控制程序梯形图（二）

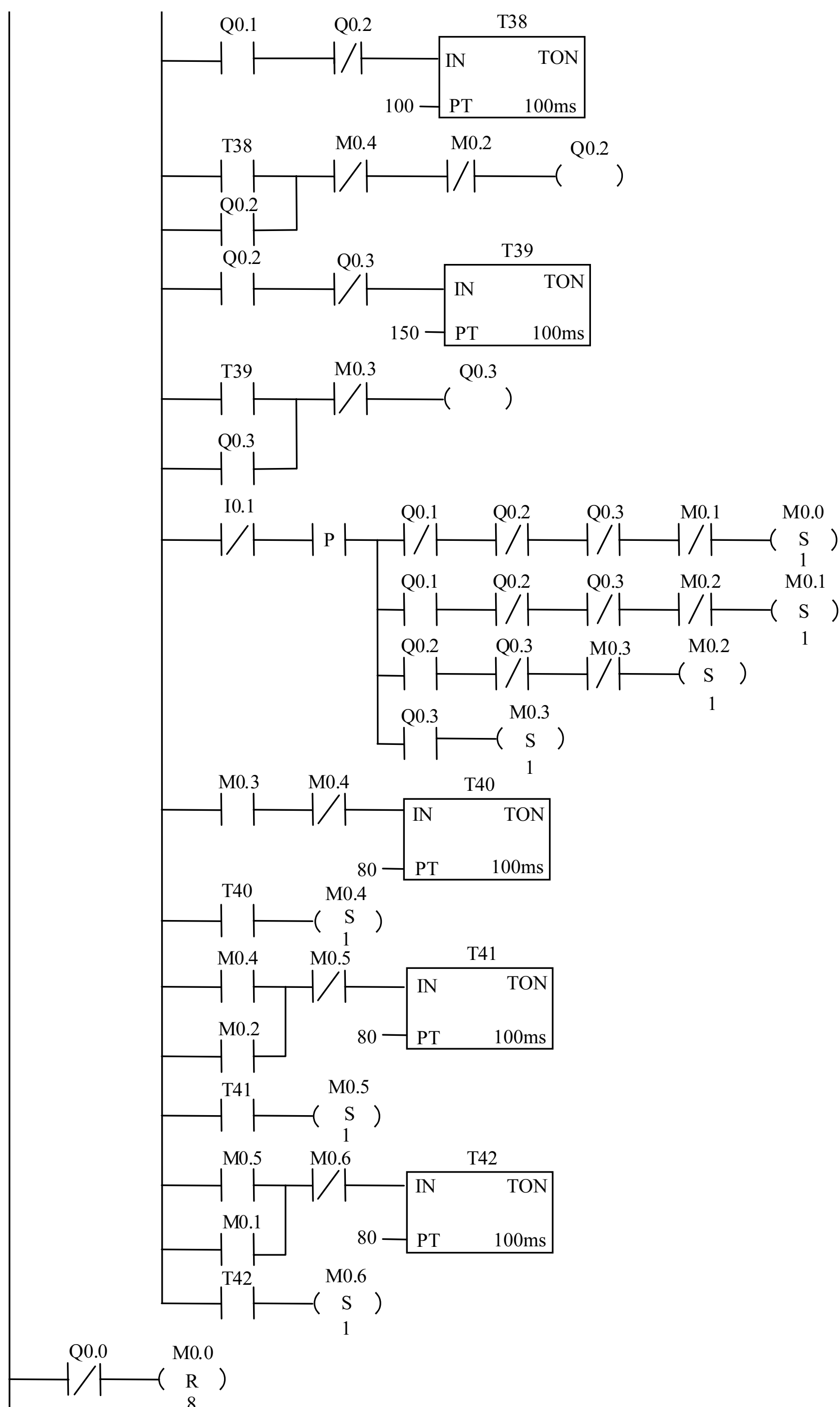


图 4-10　可重复顺起、逆停控制程序梯形图（二）（续）

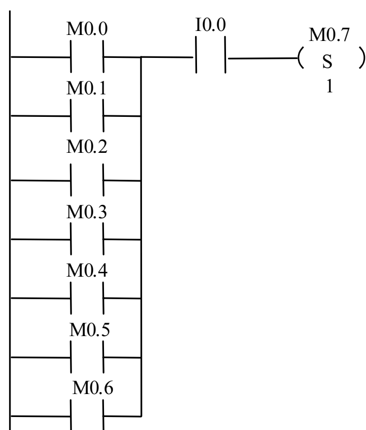


图 4-10　可重复顺起、逆停控制程序梯形图（二）（续）

3. 编程体会

编程难点在于可重复起、停，也就是说在起动过程中，按下停止按钮就会马上进入逆序停止；在停止过程中，按下起动按钮又会继续顺序起动，所有间隔时间是不能乱的。例如，逆序停止已将电机 4、电机 3 停止，再停就该是电机 2 了，就在此时又按下起动按钮 SB1，控制过程马上由停止转为起动，所以接下来的动作就是延时 10s，起动电机 3，先后顺序、时刻、间隔都是不能乱的。

方案一中只用了一个时间继电器，所有的延时都由它负责，具体时刻用了比较指令，所以程序中比较指令较多。程序中使用了上升沿脉冲指令，利用它为其前面的触点信号只 ON 一个周期的特点来抓转换点。另外，这个例子还适合用顺控指令来编，后面再专门编写。

例四、顺控指令的应用（控制要求同例三）

1. 顺序控制指令介绍

顺序控制指令属于程序控制指令的一种，在第三章第五节程序控制指令中没有具体介绍，在这里介绍该指令及其应用。

在 S7-200 PLC 中，顺序控制继电器（S0.0～S31.7）专门用于编制顺序控制程序，所以有时将顺序控制指令称为顺序控制继电器指令。

在控制系统中常常会出现控制过程具有“步”的特点，当一个转移信号发生时，当前的工作状态会有变化、还会发生两个以上的动作或动作顺序选择，这时使用顺序控制指令编程就显得简单而又容易很多。因具有“步”的特点，所以在程序执行过程中，某一时刻激活哪一步，哪一步就成为活动步，其他步都处于封闭（不活动）状态，例如，所编的程序是第一步驱动 Q0.0；第二步驱动 Q0.1；第

三步驱动 Q0.1 与 Q0.2；当程序激活第二步时只执行第二步，也就是只驱动 Q0.1，其他步的程序都不执行，更谈不上驱动输出了。

在编写程序时如何搭建“步”，就要使用顺序控制指令了。每一步都要使用 3 条指令，这 3 条指令前后呼应，顺序不能颠倒，缺一不可，组成一个固定的程序段，一个“团结战斗的阵营”。这 3 条指令是（1）段开始（SCR）；（2）段转移（SCRT）；（3）段结束（SCRE），在每一步开始时用段开始指令，接下来是在这一段要完成的控制任务，再接下来编段转移程序，也就是一旦某个转移信号出现，就要激活段转移指令，从当前步转移到段转移指令所指向的步，最后是段结束指令，它的功能是结束本步（工作段）程序的运行。表 4-1 列出这 3 条指令的形式及功能。

表 4-1 顺序控制指令的形式及功能

STL	LAD	功能	操作对象
LSCR S_bit	S_bit SCR	顺序状态开始	S（位）
SCRT S_bit	S_bit —(SCRT)	顺序状态转移	S（位）
SCRE	—(SCRE)	顺序状态结束	无

从表中看出，顺序控制指令的操作对象为顺控继电器 S，它是唯一专用于顺控指令的继电器。一个 S 位可表示一步（段）。共计 256 位（S0.0～S31.7）。

（1）段开始指令 SCR 段开始指令的功能是标记一个 SCR 段的开始，其操作数是状态继电器 Sx.y，Sx.y 是当前 SCR 段的标志位，当 Sx.y 为 1 时，允许该 SCR 段工作。

（2）段转移指令 SCRT 段转移指令的功能是将当前的 SCR 段切换到下一个 SCR，其操作数是下一个 SCR 段的标志位 Sx.y。当允许输入有效时，进行切换，即停止当前 SCR 段工作，起动下一个 SCR 段工作。

（3）段结束指令 SCRE 段结束指令的功能是标记一个 SCR 段的结束。每个 SCR 段必须使用段结束指令来表示该段的结束。在梯形图中，段开始指令以功能框的形式编程，指令名称为 SCR，段转移和段结束以线圈形式编程。

2. 顺序控制指令特点

① SCR 指令的操作数只能是顺控继电器 Sx.y；反之，S 还可当做一般继电器来使用。

② 一个顺控继电器 Sx.y 作为 SCR 段标志位，可以用于主程序、子程序或中断程序中，但是只能使用一次，不能重复使用。

③ 在一个 SCR 段中，禁止使用循环 FOR/NEXT、跳转 JMP/LBL 和条件结束 END 等指令。

④ 转移源自动复位功能：状态发生转移后，置位下一个状态的同时，自动复位原状态。

⑤ SCR 段程序能否执行取决于负责该段的（S）是否被置位，SCRE 与下一个 SCR 之间的指令逻辑不影响下一个 SCR 段程序的执行。

⑥ 在状态发生转移后，所有的 SCR 段的元器件一般也要复位，如果希望继续输出，可使用置位/复位指令。

⑦ 每一个 SCR 段都有一个 S 位编号，段与段之间编号可以不按顺序安排。

3. 应用举例

控制要求：完全按照例三的控制要求，用顺序控制指令编写程序。

程序设计：

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，运输机可重复顺序起动、逆序停止 PLC 控制接线图如图 4-8 所示。

（2）可重复顺起、逆停顺序控制程序梯形图如图 4-11 所示。

4. 编程体会

（1）与例三中的方案一、方案二相比，尽管实现的控制要求完全一样，用这种编程语言编写的程序更长一些，但语句简单很有规律性。

（2）程序刚开始的第一段 SCR，也就是本程序中 S0.0 位标志的那一段，这一段的标志位一定要用置位指令，如用一般的线圈指令也不是不可以，但出现段转移信号时，程序就要开始执行新的一段了，第一段的这个标志位一定要 OFF。说到这里可能会有些疑惑，这个标志位 S0.0 如使用了置位指令，那它是如何保证程序执行新的一段时，它会被复位？是这样的，在此刻这个标志位由置位变复位是系统程序来完成的，也就是说我们只负责搭建，PLC 自己来处理。

（3）尽管是分步执行程序，某一时刻只有活动步（段）在工作，但也不能出现双线圈现象，如第一段中出现了线圈 Q0.0，第二段中又出现了线圈 Q0.0，这是不可以的，解决的办法就是使用置位/复位指令，这样就可多次出现、通断自由了。

（4）同一编号的时间继电器只能出现一次，如在第一段中用了 T37 时间继电器，在其他段中就不能再出现了，包括它的触点都不能使用了，因这个时间继电器所处的那一段转移到另一段后，这个时间继电器就会 OFF，它的触点在其他段中是没有使用价值的。

（5）同一编号的计数器也只能出现一次，不过计数器与时间继电器是不同的，体现在计数器的状态是可保持的，尽管计数器本身所处的步（段）已封闭，计数器当前所计的次数及触点状态是可以用在其他步（段）中的，直到计数器所处的步（段）又被激活，并且复位信号已出现，这时计数器才回到原始状态。

（6）某一时刻可以同时激活两个及以上步（段），这就是选择、合并、并行、分支，是顺序控制指令程序结构的特点之一。

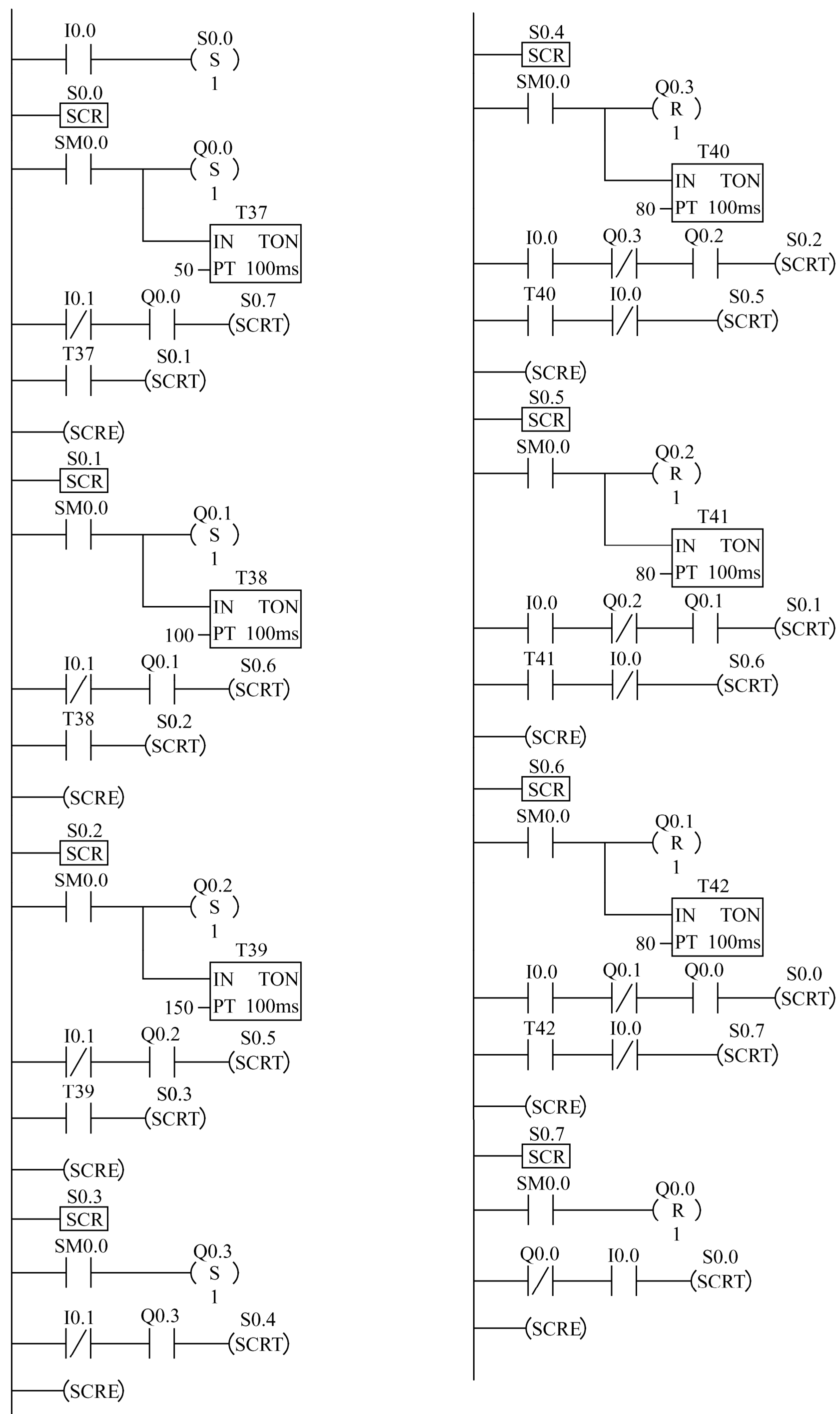


图 4-11 可重复顺起、逆停顺序控制程序梯形图

例五、运料小车自动装料、卸料控制

1. 控制要求

小车可在 A、B 两地分别起动。小车起动后自动回 A 地等待装料，若 A 处有物料，装料电磁阀动作，30s 后装料结束，小车自动向 B 地运行。到达 B 地后，停止 1min 等待卸料，然后再自动返回 A 地，如此往复。若 A 处物料已空，则小车返回 A 点后停止运行。小车在运行过程中，均可用手动开关令其停车。再次起动后，小车重复以上过程。小车在前进或后退过程中，分别用指示灯显示行进的方向。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，运料小车运行示意图如图 4-12 所示，运料小车的 PLC 控制接线图如图 4-13 所示。

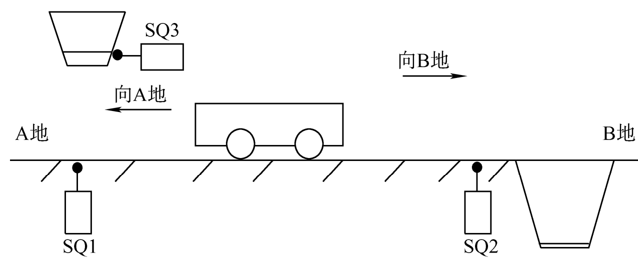


图 4-12　小车运行示意图

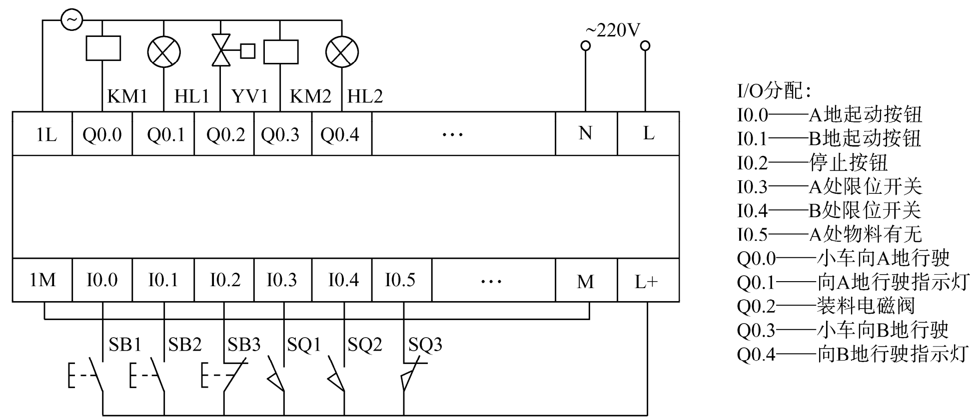


图 4-13　运料小车的 PLC 控制接线图

（2）控制程序梯形图如图 4-14 所示。

图 4-14 控制程序梯形图

例六、四组抢答器程序设计

1. 控制要求

设有四组抢答器，有 4 位选手，一位主持人，主持人有一个开始答题按钮，一个系统复位按钮。如果主持人按下开始答题按钮后，4 位选手开始抢答，抢先按下按钮的选手的正常抢答指示灯亮，同时选手序号在数码管上显示，其他选手按钮不起作用。如果主持人未按下开始答题按钮，就有选手抢答，则认为犯规，犯规指示灯亮并闪烁，同时选手序号在数码管上显示，其他选手按钮不起作用。当主持人按下开始答题按钮，时间开始倒计时，在 10s 内仍无选手抢答，则系统超时指示灯亮，此后不能再有选手抢答。所有各种情况，只要主持人按下系统复位按钮后，系统回到初始状态。

此题接线时一定要注意将选手抢答按钮的常闭触点串入其他选手的常开触点中，也就是说每个按钮应具备一组常开触点三组常闭触点，这样就更加保证了抢答的合理性。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，抢答器示意图如图 4-15 所示，抢答器 PLC 控制接线图如图 4-16 所示。

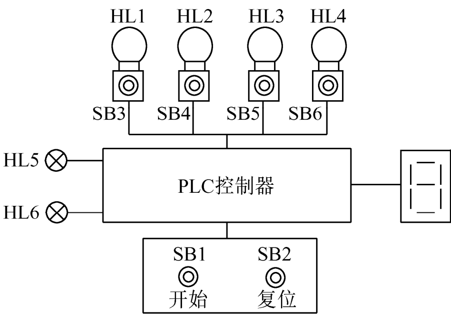


图 4-15　抢答器示意图

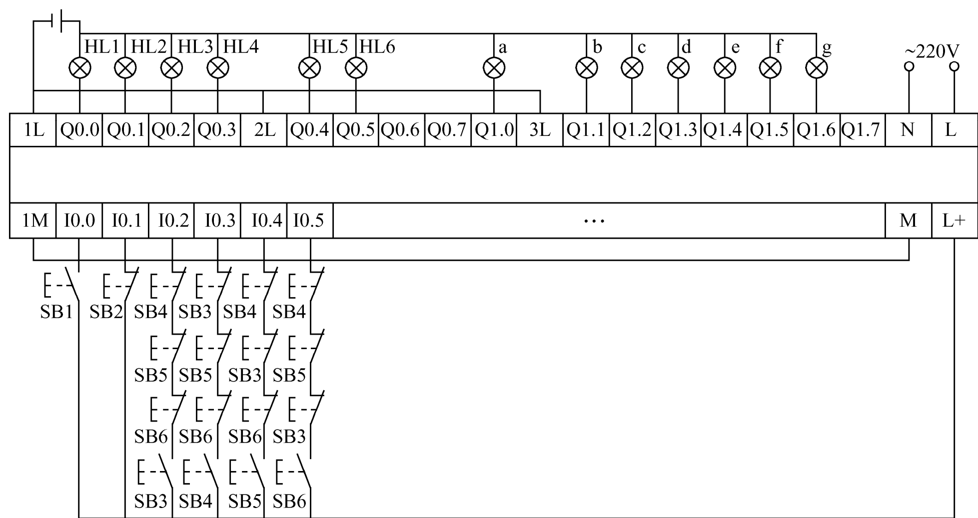


图 4-16　抢答器 PLC 控制接线图

在图 4-16 中，主持人开始答题按钮（SB1）；主持人复位按钮（SB2）；I0.2 至 I0.5 四位选手抢答按钮（SB3～SB6）；Q0.0 至 Q0.3 对应四位选手指示灯（HL1～HL4）；Q0.4 没人抢答灯（HL5）；Q0.5 犯规指示灯（HL6）。

（2）控制程序梯形图如图 4-17 所示。

（3）程序的语句表及注释如下：

```
Network 1                    // 程序注释
LD      I0.0
AN      M0.3
```

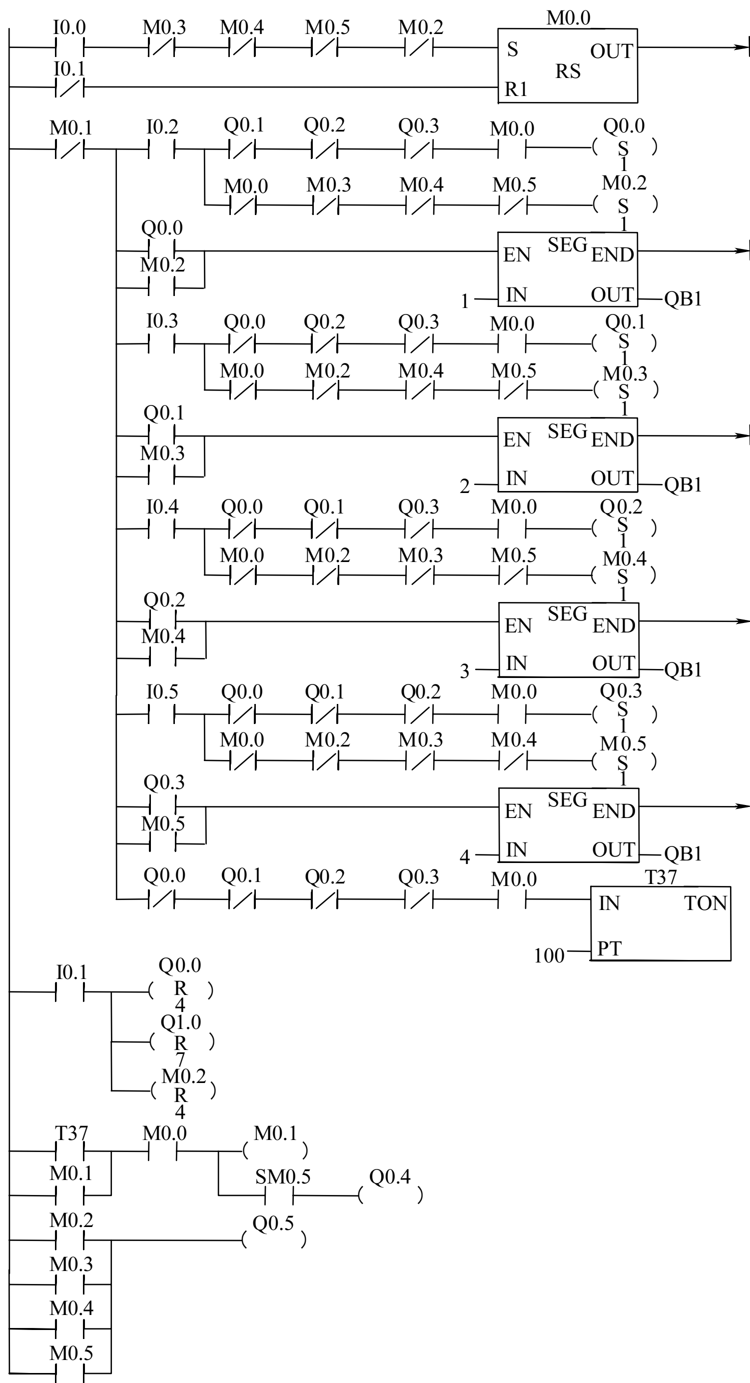


图 4-17 抢答器控制程序梯形图

```
AN      M0.4
AN      M0.5
AN      M0.2
LDN     I0.1
NOT
LPS
A       M0.0
=       M0.0            // 主持人控制抢答或复位
LPP
ALD
O       M0.0
=       M0.0
Network 2
LDN     M0.1
LPS
A       I0.2
LPS
AN      Q0.1
AN      Q0.2
AN      Q0.3
A       M0.0
S       Q0.0, 1         // 1 号选手正常抢答
LPP
AN      M0.0
AN      M0.3
AN      M0.4
AN      M0.5
S       M0.2, 1         // 1 号选手犯规抢答
LRD
LD      Q0.0
O       M0.2
ALD
SEG     1, QB1          // 1 号选手数码显示
LRD
A       I0.3
```

```
LPS
AN      Q0.0
AN      Q0.2
AN      Q0.3
A       M0.0
S       Q0.1, 1         //  2 号选手正常抢答
LPP
AN      M0.0
AN      M0.2
AN      M0.4
AN      M0.5
S       M0.3, 1         //  2 号选手犯规抢答
LRD
LD      Q0.1
O       M0.3
ALD
SEG     2, QB1          //  2 号选手数码显示
LRD
A       I0.4
LPS
AN      Q0.0
AN      Q0.1
AN      Q0.3
A       M0.0
S       Q0.2, 1         //  3 号选手正常抢答
LPP
AN      M0.0
AN      M0.2
AN      M0.3
AN      M0.5
S       M0.4, 1         //  3 号选手犯规抢答
LRD
LD      Q0.2
O       M0.4
ALD
```

```
SEG     3, QB1        // 3 号选手数码显示
LRD
A       I0.5
LPS
AN      Q0.0
AN      Q0.1
AN      Q0.2
A       M0.0
S       Q0.3, 1       // 4 号选手正常抢答
LPP
AN      M0.0
AN      M0.2
AN      M0.3
AN      M0.4
S       M0.5, 1       // 4 号选手犯规抢答
LRD
LD      Q0.3
O       M0.5
ALD
SEG     4, QB1        // 4 号选手数码显示
LPP
AN      Q0.0
AN      Q0.1
AN      Q0.2
AN      Q0.3
A       M0.0
TON     T37, 100      // 无人抢答倒计时
Network 3
LDN     I0.1
R       Q0.0, 4       // 按复位钮，所有点都复位
R       Q1.0, 7
R       M0.2, 4
Network 4
LD      T37
O       M0.1
```

```
A       M0.0
=       M0.1
=       Q0.4                // 超时无人抢答指示灯亮
Network 5
LD      M0.2
O       M0.3
O       M0.4
O       M0.5
A       SM0.5
=       Q0.5                // 犯规指示灯亮并闪烁
```

例七、单按钮控制彩灯循环

1. 控制要求

用 1 个按钮控制彩灯循环，第一次按下起动循环，第二次按下停止循环。用另一个按钮控制循环方向，第一次按下左循环，第二次按下右循环，由此交替，假设初始状态为 00000101，循环周期为 1s。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，彩灯循环 PLC 控制接线图如图 4-18 所示。

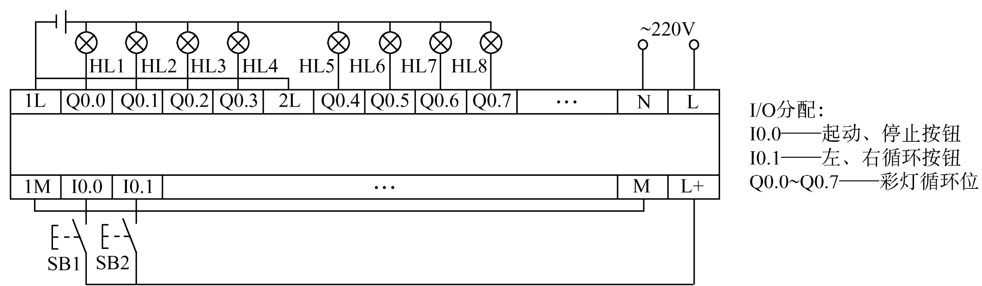


图 4-18　彩灯循环 PLC 控制接线图

（2）控制程序梯形图如图 4-19 所示。

（3）程序的语句表及注释如下：

```
Network 1               // 程序注释
LD      SM0.1
MOVB    16#05, QB0      // 将 00000101 送入 QB0 中
Network 2
```

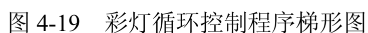


图 4-19　彩灯循环控制程序梯形图

```
LD        I0.0
EU
LD        I0.0
EU
A         M0.0
NOT
LPS
A         M0.0
=         M0.0
LPP
ALD
O         M0.0
=         M0.0              // 控制起动与停止
Network 3
LD        M0.0
```

```
AN       T37
TON      T37, 10          // 定循环周期
Network 4
LD       I0.1
EU
LD       I0.1
EU
A        M0.1
NOT
LPS
A        M0.1
=        M0.1
LPP
ALD
O        M0.1
=        M0.1             // 决定左移或右移
Network 5
LD       T37
LPS
A        M0.1
RLB      QB0, 1           // 左移
LPP
AN       M0.1
RRB      QB0, 1           // 右移
```

例八、洗衣机自动控制程序设计

1. 控制要求

初始状态：没有任何输出信号，洗衣机处于静止状态。

合上洗衣机起动开关 SA1。①开始往洗衣机里注水，进水电磁阀 KV1 工作，待水位到达水位满位置时，水位开关 SL2 闭合，此时低水位位置上的水位开关 SL1 肯定也是闭合的，停止进水，KV1 断电，洗衣机开始正转，正转 10s 后，停止 5s，洗衣机反转，反转 10s 后，停止 5s。如此正反转再重复 2 次，共 3 次，停止转动。②开始排水，排水电磁阀 KV2 工作，待水位下降到 SL1 开关以下时，停止排水，KV2 断电。洗衣机又重新进水，重复①的工作过程，然后再排水，再重复①，总计重复 2 次①的过程，相当于①的工作过程 3 次，排水 3 次。③第 3 次排水后，

待水位下降到 SL1 开关以下时，停止排水，KV2 断电。洗衣机进入脱水工作段，脱水共需 5s，然后全部工作过程结束。④无论何时合上洗衣机停止开关 SA2，停止当前操作，回到初始状态。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，如图 4-20 为洗衣机工作示意图，洗衣机自动洗衣 PLC 控制接线图如图 4-21 所示。

（2）控制程序梯形图如图 4-22 所示。

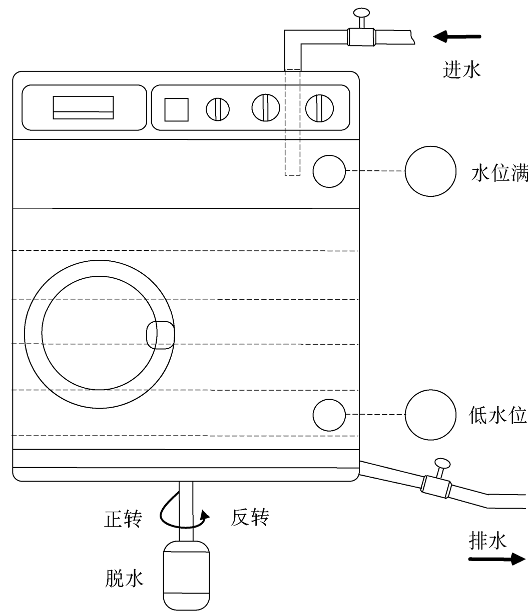


图 4-20　洗衣机工作示意图

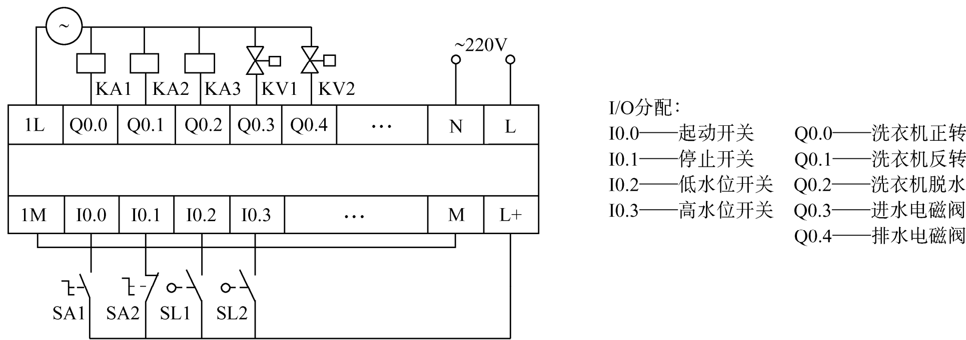
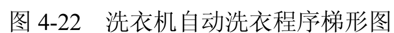


图 4-21　洗衣机自动洗衣 PLC 控制接线图

（3）程序的语句表及注释如下：

I0.0 P S0.0 (S) 1
S0.0 SCR
I0.0 Q0.3 ( )
I0.0 I0.2 I0.3 S0.1 (SCRT)
(SCRE)
S0.1 SCR
I0.0 Q0.0 ( )
T37 IN TON 100 PT 100ms
T37 S0.2 (SCRT)
(SCRE)
S0.2 SCR
I0.0 T38 IN TON 50 PT 100ms
T38 S0.3 (SCRT)
(SCRE)
S0.3 SCR
I0.0 Q0.1 ( )
T39 IN TON 100 PT 100ms
T39 S0.4 (SCRT)
(SCRE)
S0.4 SCR
I0.0 T40 IN TON 50 PT 100ms

T40 C1 CU CTU
C1 R
3 PV
T40 C1 S0.1 (SCRT)
T40 C1 S0.5 (SCRT)
(SCRE)
S0.5 SCR
I0.0 Q0.4 ( )
I0.0 I0.2 I0.3 C2 CU CTU
C2 R
3 PV
C2 I0.2 I0.3 S0.0 (SCRT)
C2 I0.2 I0.3 S0.6 (SCRT)
(SCRE)
S0.6 SCR
I0.0 Q0.2 ( )
T41 IN TON 50 PT 100ms
T41 S0.7 (SCRT)
(SCRE)
I0.1 Q0.0 (R) 8
S0.0 (R) 8


图 4-22 洗衣机自动洗衣程序梯形图

```
Network 1                       // 程序注释
LD       I0.0                   // 开启工作按钮
EU
S        S0.0, 1                // 起动顺序控制程序的第一步
Network 2
LSCR     S0.0
Network 3
LD       I0.0
=        Q0.3 门                // 起动进水电磁阀
Network 4
LD       I0.0
A        I0.2
A        I0.3
SCRT     S0.1                   // 当水位已超最高水位点，出现步转移
Network 5
SCRE
Network 6
LSCR     S0.1
Network 7
LD       I0.0
=        Q0.0                   // 新的一步是洗衣机开始正转
TON      T37, +100              // 正转 10s
Network 8
LD       T37
SCRT     S0.2                   // 10s 后出现步转移
Network 9
SCRE
Network 10
LSCR     S0.2
Network 11
LD       I0.0
TON      T38, +50               // 新的一步是洗衣机静止 5s
Network 12
LD       T38
SCRT     S0.3                   // 5s 后出现步转移
```

```
Network 13
SCRE
Network 14
LSCR    S0.3
Network 15
LD      I0.0
=       Q0.1              // 新的一步是洗衣机开始反转
TON     T39, +100         // 反转 10s
Network 16
LD      T39
SCRT    S0.4              // 10s 后出现步转移
Network 17
SCRE
Network 18
LSCR    S0.4
Network 19
LD      I0.0
TON     T40, +50          // 新的一步是洗衣机静止 5s
Network 20
LD      T40
LD      C1
CTU     C1, 3             // 5s 后给计数器计数一次共计 3 次
Network 21
LD      T40
AN      C1
SCRT    S0.1              // 3 次没到，就往回返
Network 22
LD      T40
A       C1
SCRT    S0.5              // 3 次到了，就转移到新的一步
Network 23
SCRE
Network 24
LSCR    S0.5
Network 25
```

```
LD        I0.0
=         Q0.4                // 新的一步是排水
Network 26
LD        I0.0
AN        I0.2
AN        I0.3
LD        C2
CTU       C2, 3               // 排水过程计数，总共应排 3 次
Network 27
LDN       C2
AN        I0.2
AN        I0.3
SCRT      S0.0                // 如没到 3 次，就回到最初步，重新加水继续洗
Network 28
LD        C2
AN        I0.2
AN        I0.3
SCRT      S0.6                // 如已到 3 次，就出现步转移
Network 29
SCRE
Network 30
LSCR      S0.6
Network 31
LD        I0.0
=         Q0.2                // 新的一步是脱水
TON       T41, +50            // 脱水共用 5s
Network 32
LD        T41
SCRT      S0.7                // 5s 后全部洗衣过程结束
Network 33
SCRE
Network 34
LDN       I0.1        // 在洗衣过程中什么时候按下停止按钮，都回到初始状态
R         S0.0, 8
R         Q0.0, 8
```

例九、高速计数器应用程序

1. 控制要求

包装箱用传送带输送，当箱体到达检测传感器 A 时，开始计数。计数到 2000 个脉冲时，箱体刚好到达封箱机下进行封箱，此时传送带并没有停下，而是继续运转。则在封箱过程中，箱体还在前行。假设封箱过程共用 300 个脉冲，然后封箱机停止工作。继续前行，当计数脉冲又累加到 1500 个码时，开始喷码，喷码机开始工作，假设喷码机共用 5s 进行喷码，喷码结束后，整个工作过程结束。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，通过本题应了解和掌握高速计数器的编程大致需哪些初始化指令，定好模式后，按照所开通的高速计数器的号数，把旋转编码器与 PLC 输入端之间的导线接好。本题的模式是 9，开通的是 0 号高速计数器，所以将旋转编码器的 A、B 相分别接到 I0.0 及 I0.1 即可。箱体输送过程示意图如图 4-23 所示，控制系统 PLC 接线图如图 4-24 所示。

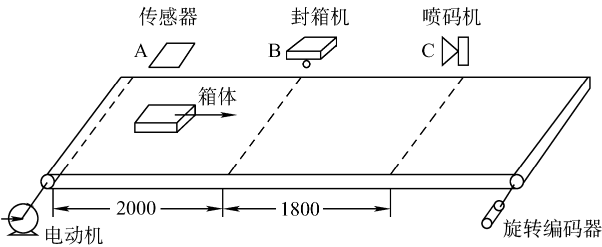


图 4-23 箱体输送过程示意图

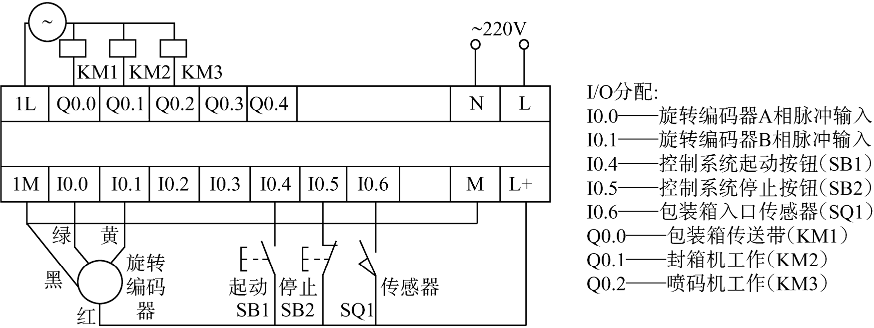


图 4-24 箱体输送控制系统 PLC 接线图

（2）控制程序梯形图如图 4-25 所示。

（3）程序的语句表及注释如下：

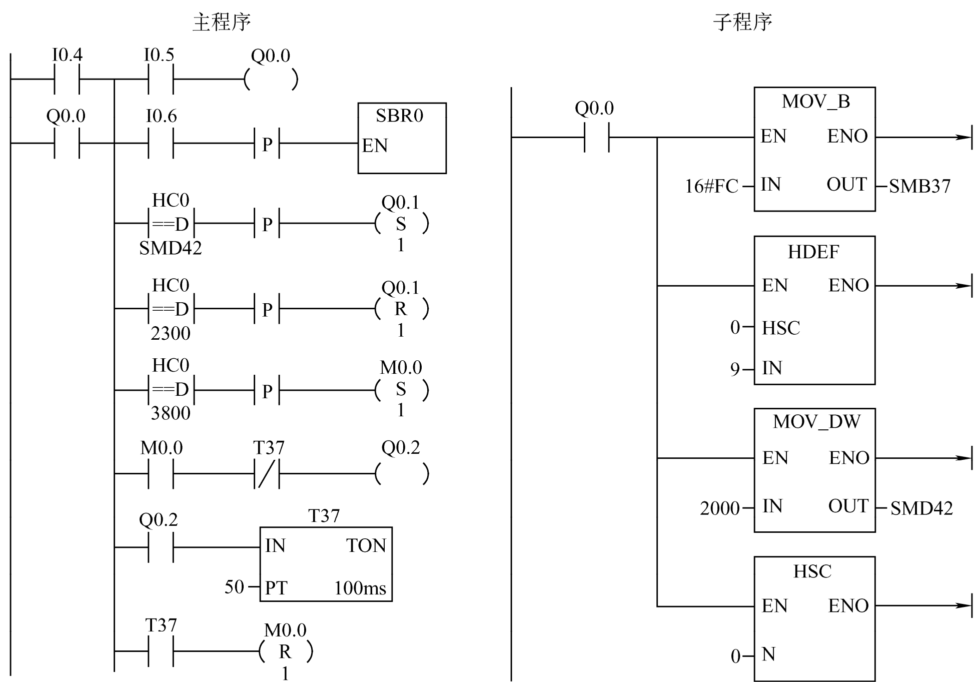


图 4-25　箱体输送控制程序梯形图

主程序：

```
Network 1               // 程序注释
LD      I0.4
O       Q0.0
LPS
A       I0.5
=       Q0.0            // 起动传送带电动机
LRD
A       I0.6
EU
CALL    SBR0            // 转入子程序，初始化设置
LRD
AD=     HC0, SMD42
EU
S       Q0.1, 1         // 当计数脉冲个数与设定值相等时，开始封箱
LRD
```

```
AD=     HC0, 2300
EU
R       Q0.1, 1          // 停止封箱机工作
LRD
AD=     HC0, 3800
EU
S       M0.0, 1          // 当计数到 3800 个脉冲时，开始喷码工序
LRD
A       M0.0
AN      T37
=       Q0.2             // 起动喷码机
LRD
A       Q0.2
TON     T37, 50          // 喷码过程所用时间
LPP
A       T37
R       M0.0, 1          // 喷码过程结束
子程序：
Network 1
LD      Q0.0
MOVB    16#FC, SMB37     // 给高速计数器定工作状态
HDEF    0, 9             // 决定使用 0 号计数器，用第 9 个模式
MOVD    2000, SMD42      // 给高速计数器设定值
HSC     0                // 开通 0 号高速计数器
```

3. 编程体会

高速计数器功能指令是当今 PLC 的强功能应用指令之一，常常利用它控制位移量和速度等。一旦编好程序，将某个 PLC 的输入点作为高速计数器信号输入点，此点输入信号就可“畅通无阻”，不受扫描周期的限制，最高频率可达 30kHz。在硬件接线方面，现大多数都是采用旋转编码器作为 PLC 的高频输入信号，所以只要将编码器的电源线、脉冲信号线与 PLC 的相应位置接好即可。编程方面，有几个功能指令是必须编的：①定控制字节；②定工作模式；③定高速计数器号别，其次是定设定值及当前值（可有可无）。在本例中将这几条都放在子程序中，初始化激活即可，决不可以每次都去扫描这几条，那样的话，高速计数器就无法运行计数了。具体利用高速计数器的码数干什么那就看控制要求了，因旋转编码器送进来的脉冲码远远高于 PLC 的扫描速度，所以要抓一个具体数据要跟上 P 指令。

例十、基于高速计数器的电梯层高自学习及层标显示控制

1. 控制要求

利用高速计数器确定楼层层间距，高速计数器的输入信号由 A/B 正交型旋转编码器提供。方法是把电梯停在基站的平层位置，压住下行强迫换速开关（SQ1），拨动专用的自学习开关（QS）使电梯以极慢的速度上行。现假设电梯共有 5 层，每到一层平层处都有一个门区信号（SQ3），利用此开关将高速计数器的脉冲数截住既为层间距。然后放到对应的寄存器中，电梯继续运行直到最顶层，压住上行强迫换速开关（SQ2）时电梯停止，这时层间距都已寄存，自学习过程结束。

只用两个按钮作为上、下运行起动，起动后电梯可在 1 层到 5 层之间上下运行，层标也随之从 1 计到 5，碰到限位开关自动返回由此往复，按停止按钮可停止。

层间距定好后就可以进行模拟运行，不考虑任何交通呼梯信号及减速停靠，电梯只是从基站到顶层直线运行，途中不停靠只是有相应的层标显示。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配。进一步了解高速计数器功能，在电梯控制系统中，旋转编码器的脉冲一方面给变频器用来控制速度，另一方面给 PLC 用来表示位移量，从而决定停靠时刻及层标号等。控制系统 PLC 接线图如图 4-26 所示。

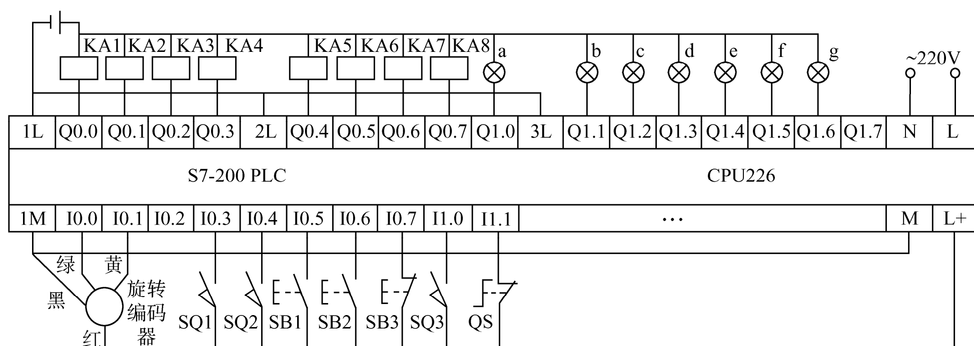


图 4-26　控制系统 PLC 接线图

I：I0.0 I0.1 I0.2 给高速计数器　I0.3—下行强迫换速开关（SQ1）　I1.0—门区光电开关（SQ3）　I0.4—上行强迫换速开关（SQ2）　I1.1—自学习开关（QS）　I0.5　I0.6 正反向起动按钮（SB1、SB2）　I0.7 停止按钮（SB3）

Q：Q0.0 电梯上行（KA1）Q0.1—Q0.5 对应 1 至 5 层的输出信号（KA2—KA6）Q0.6 电梯下行（KA7）　Q0.7 井道自学习速度（KA8）　Q1.0 到 Q1.7 七段数码管层标显示（a　b　c　d　e　f　g）

层间距寄存器分配：VD100（1 层到 2 层间）　　VD200（2 层到 3 层间）

VD300（3 层到 4 层间） VD400（4 层到 5 层间）

（2）控制程序梯形图如图 4-27 所示。

主程序：

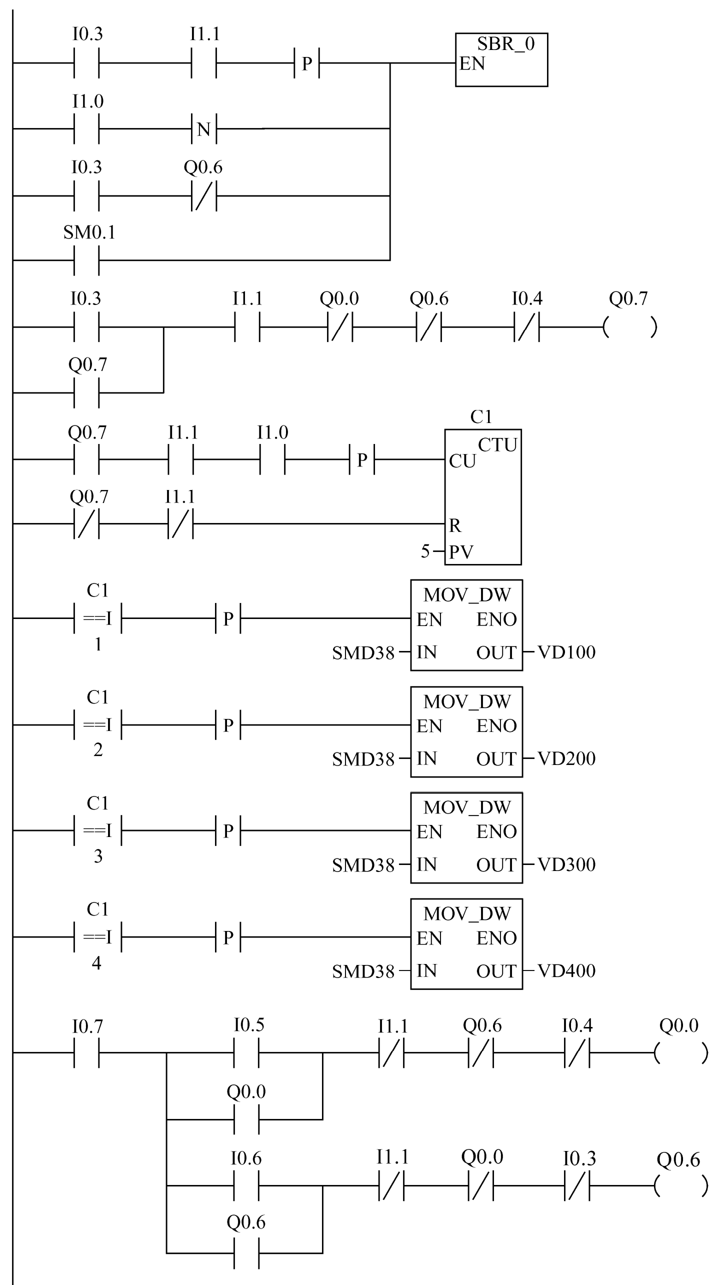


图 4-27 电梯层标控制程序梯形图

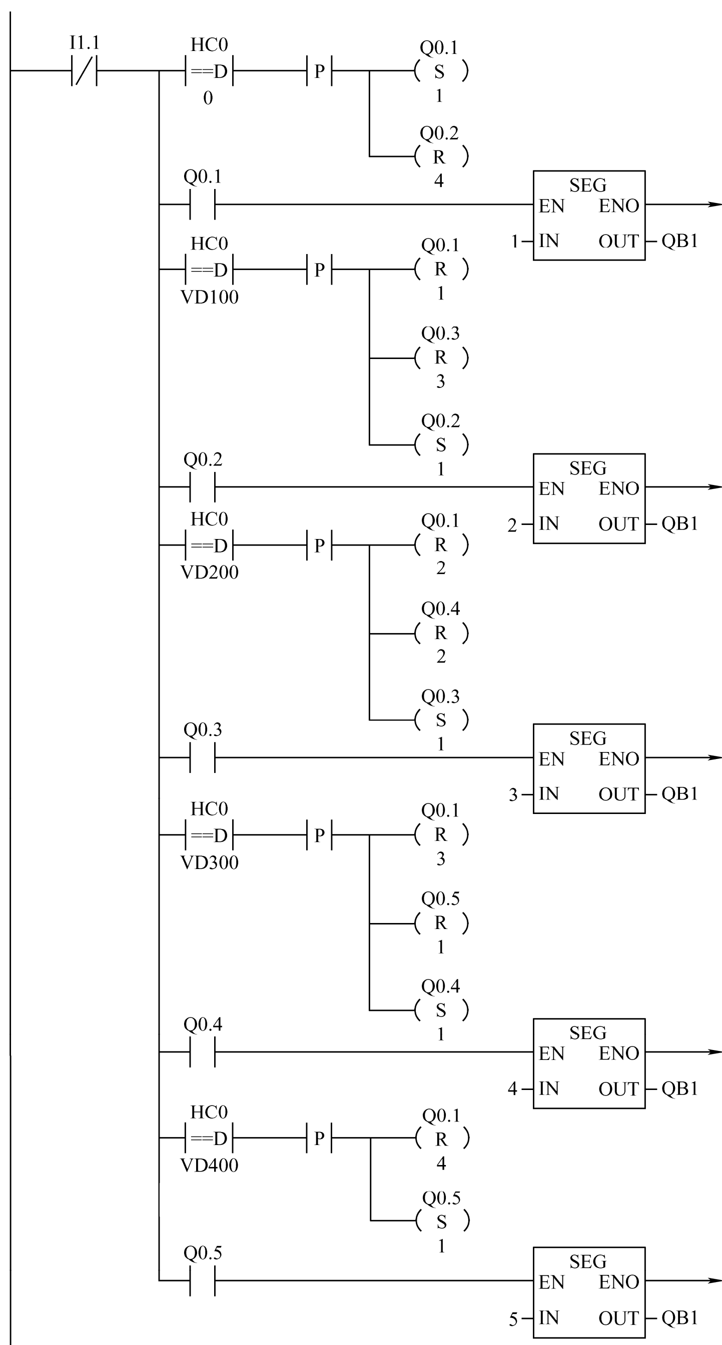


图 4-27　电梯层标控制程序梯形图（续）

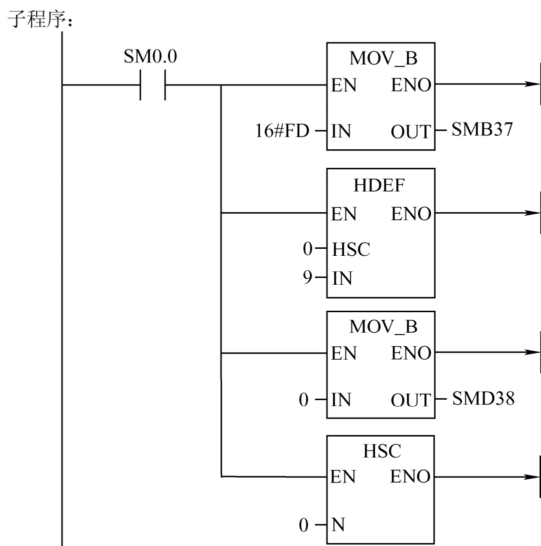


图 4-27　电梯层标控制程序梯形图（续）

（3）程序的语句表及注释如下：

```
主程序（OB1）            // 程序注释
Network 1               // 进入高速计数器初始化程序段
LD      I0.3            // 电梯在基站压住下行强迫换速开关
A       I1.1            // 打开井道自学习开关
EU
LD      I1.0
ED                      // 门区光电开关的下降沿使高速计数器清零
OLD
O       SM0.1           // 刚打开机器时高速计数器清零
LD      I0.3
AN      Q0.6
OLD                     // 正常运行每到基站时将使高速计数器清零
CALL    SBR0            // 调子程序（高速计数器初始化）
Network 2               // 井道自学习程序段
LD      I0.3            // 电梯在基站压住下行强迫换速开关
O       Q0.7
A       I1.1            // 打开井道自学习开关
AN      Q0.0
AN      Q0.6            // 不能有正反向正常运行
```

```
AN      I0.4             // 到上行强迫换速开关必须停止自学习
=       Q0.7             // 井道自学习专用速度
Network 3
LD      Q0.7
A       I1.1
A       I1.0
EU                       // 井道自学习过程中每到一层增计数器就累加一次
LDN     Q0.7
AN      I1.1             // 如没有井道自学习操作，此计数器不工作
CTU     C1, 5            // 井道自学习过程中计算层间距的计数器
Network 4
LDW=    C1, 1            // 当普通计数器的计数为 1 时，电梯就到了 2 层
EU
MOVD    SMD38, VD100     // 将 1 至 2 层之间的数码保存在变量寄存器中
Network 5
LDW=    C1, 2            // 当普通计数器的计数为 2 时，电梯就到了 3 层
EU
MOVD    SMD38, VD200     // 将 2 至 3 层之间的数码保存在变量寄存器中
Network 6
LDW=    C1, 3            // 当普通计数器的计数为 3 时，电梯就到了 4 层
EU
MOVD    SMD38, VD300     // 将 3 至 4 层之间的数码保存在变量寄存器中
Network 7
LDW=    C1, 4            // 当普通计数器的计数为 4 时，电梯就到了 5 层
EU
MOVD    SMD38, VD400     // 将 4 至 5 层之间的数码保存在变量寄存器中
Network 8                // 以下是井道自学习结束后的上下运行程序段
LD      I0.7             // 停止按钮
LPS
LD      I0.5             // 上行起动按钮
O       Q0.0
ALD
AN      I0.4
AN      I1.1
AN      Q0.6
```

```
=       Q0.0            // 电梯上行运行输出信号
LPP
LD      I0.6            // 下行起动按钮
O       Q0.6
ALD
AN      I0.3
AN      I1.1
AN      Q0.0
=       Q0.6            // 电梯下行运行输出信号
Network 9
LDN     I1.1
LPS
AD=     HC0, 0          // 当高速计数器的数码为 0 时
EU
S       Q0.1, 1
R       Q0.2, 4
LRD
A       Q0.1
SEG     1, QB1          // 层标显示为 1 层
LRD
AD=     HC0, VD100      // 当高速计数器的数码等于 1 至 2 层层间距时
EU
S       Q0.2, 1
R       Q0.1, 1
R       Q0.3, 3
LRD
A       Q0.2
SEG     2, QB1          // 层标显示为 2 层
LRD
AD=     HC0, VD200      // 当高速计数器的数码等于 2 至 3 层层间距时
EU
S       Q0.3, 1
R       Q0.1, 2
R       Q0.4, 2
LRD
```

```
A       Q0.3
SEG     3, QB1          // 层标显示为 3 层
LRD
AD=     HC0, VD300      // 当高速计数器的数码等于 3 至 4 层层间距时
EU
S       Q0.4, 1
R       Q0.1, 3
R       Q0.5, 1
LRD
A       Q0.4
SEG     4, QB1          // 层标显示为 4 层
LRD
AD=     HC0, VD400      // 当高速计数器的数码等于 4 至 5 层层间距时
EU
S       Q0.5, 1
R       Q0.1, 4
LPP
A       Q0.5
SEG     5, QB1          // 层标显示为 5 层
子程序（SBR_0）          // 以下是给高速计数器初始化的子程序
LD      SM0.0
MOVB    16#FD, SMB37    // 给高速计数器定工作状态
HDEF    0, 9            // 给高速计数器定模式
MOVD    0, SMD38        // 给高速计数器当前值存储器清零
HSC     0               // 开通 0 号高速计数器
```

3. 编程体会

本例的控制要求只是一个实验室模拟的电梯控制系统，实际工程还要比这复杂得多，目的是进一步介绍高速计数器的应用场合。在实际电梯控制系统，交通信号随机性很强，一部高层电梯也许要有 2～3 个运行速度，在编写电梯减速停靠程序时都要考虑进去，本例只是一个大致的框架，距实际还差很远。

例十一、人行道交通信号灯控制

1. 控制要求

这是一条公路与人行横道之间的信号灯顺序控制，没有人横穿公路时，公路绿灯与人行道红灯始终都是亮的，当有人需过路时按路边设有的按钮（两侧均设）

SB1 或 SB2，15s 后公路绿灯灭、黄灯亮，再过 10s 黄灯灭、红灯亮，再等 5s 人行道红灯灭、绿灯亮，绿灯亮 10s 后又闪烁 4. 5s，然后人行道红灯又亮了，再过 5s 公路红灯灭、绿灯亮，在整个过程中按路边的按钮是不起作用的，只有当整个过程结束后也就是公路绿灯与人行道红灯同时亮时，再按按钮才起作用。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配。交通信号灯时序图如图 4-28 所示，控制系统 PLC 接线图如图 4-29 所示。

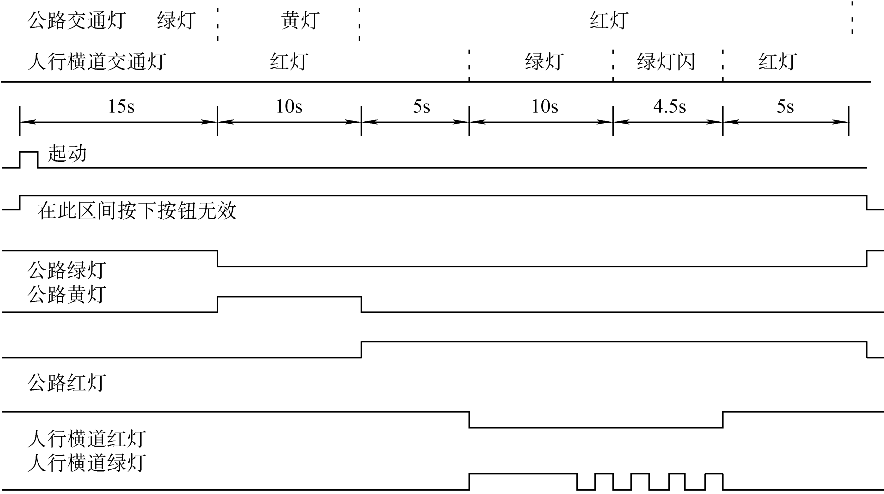


图 4-28　交通信号灯时序图

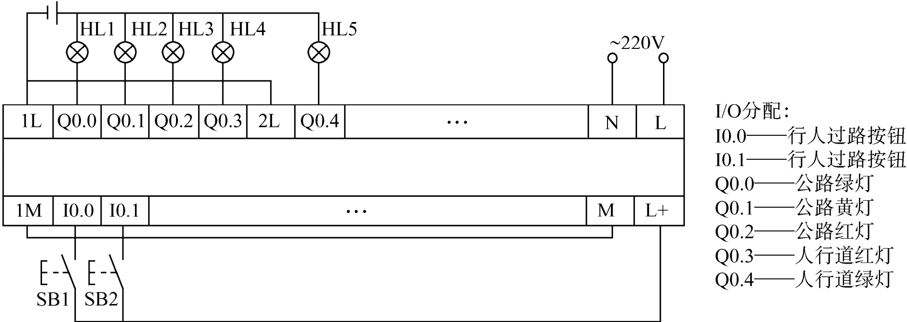


图 4-29　交通信号灯 PLC 控制接线图

（2）控制程序梯形图如图 4-30 所示。

（3）程序的语句表及注释如下：

```
Network 1                    //  程序注释
LD        SM0.0
LPS
```

```
AN      M0.1
=       Q0.0            // 公路绿灯
LRD
LD      I0.0
O       I0.1
ALD
EU
S       M0.0, 1         // 建立行人过路信号
LRD
A       M0.0
```

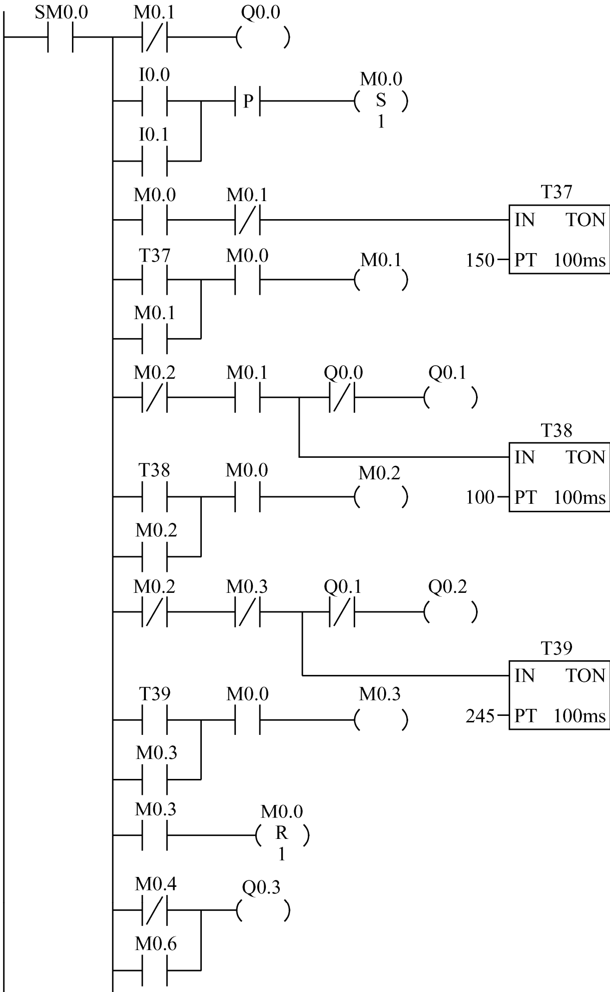


图 4-30　交通信号灯控制程序梯形图

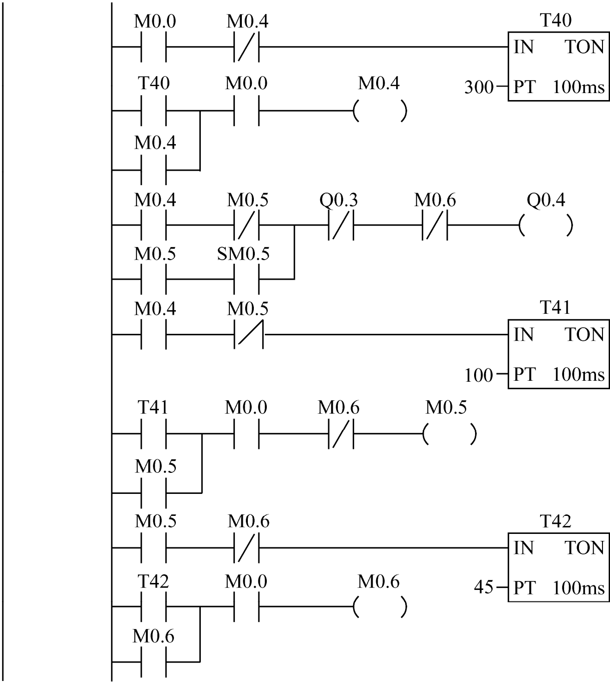


图 4-30　交通信号灯控制程序梯形图（续）

```
AN      M0.1
TON     T37, 150        // 建立行人过路信号后延时 15s 公路绿灯灭
LRD
LD      T37
O       M0.1
ALD
A       M0.0
=       M0.1            // 断开公路绿灯
LRD
AN      M0.2
A       M0.1
LPS
AN      Q0.0
=       Q0.1            // 公路黄灯
LPP
TON     T38, 100        // 公路黄灯点亮时间
LRD
```

```
LD        T38
O         M0.2
ALD
A         M0.0
=         M0.2              // 断开公路黄灯点亮公路红灯
LRD
A         M0.2
AN        M0.3
LPS
AN        Q0.1
=         Q0.2              // 公路红灯
LPP
TON       T39, 245          // 控制公路红灯点亮的时间
LRD
LD        T39
O         M0.3
ALD
A         M0.0
=         M0.3              // 控制信号灯时序周期
LPP
A         M0.3
R         M0.0, 1           // 信号灯时序周期后使按钮有效
Network 2                   // 程序注解
LD        SM0.0
LPS
LDN       M0.4
O         M0.6
ALD
=         Q0.3              // 人行道红灯
LRD
A         M0.0
AN        M0.4
TON       T40, 300          // 按钮按下后延时 30s 人行道红灯灭绿灯亮
LRD
LD        T40
```

```
O       M0.4
ALD
A       M0.0
=       M0.4            // 控制人行道红灯灭绿灯亮
LRD
LD      M0.4
AN      M0.5
LD      M0.5
A       SM0.5
OLD
ALD
AN      Q0.3
AN      M0.6
=       Q0.4            // 人行道绿灯
LRD
A       M0.4
AN      M0.5
TON     T41, 100        // 人行道绿灯亮后 10s 变闪亮
LRD
LD      T41
O       M0.5
ALD
A       M0.0
AN      M0.6
=       M0.5            // 控制人行道绿灯常亮变闪亮
LRD
A       M0.5
AN      M0.6
TON     T42, 45         // 人行道绿灯闪亮时间
LPP
LD      T42
O       M0.6
ALD
A       M0.0
=       M0.6            // 人行道信号灯时序周期结束
```

例十二、步进电动机运转控制

1. 控制要求

四相八拍步进电动机接线原理图如图 4-31 所示。图中接线端 A、B、C、D 为脉冲电源输入端，E、F 为公共端。其控制要求为：

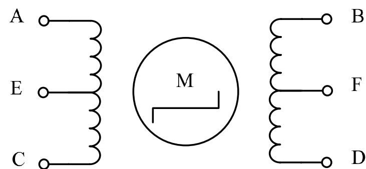


图 4-31　步进电动机接线原理图

（1）按下电动机正向起动按钮，顺序步为 A→AB→B→BC→C→CD→D→DA→A，此时电动机为正向转动。

（2）按下电动机反向起动按钮，顺序步为 A→AD→D→DC→C→CB→B→BA→A，此时电动机为反向转动。

（3）慢速为 1 步/s；快速为 10 步/s。

2. 程序设计

（1）步进电动机的 PLC 控制接线图如图 4-32 所示。

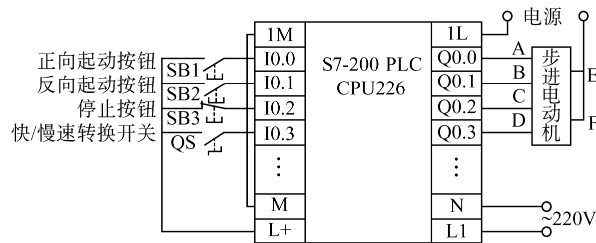


图 4-32　步进电动机 PLC 控制接线图

（2）语句表程序及注释如下：

```
主程序                  // 程序注释
Network 1               // 决定步进电动机转动方向及步进速度程序段
LD      I0.2            // 停止按钮
LPS
LD      I0.0            // 正转起动
O       M1.0
ALD
AN      I0.1
=       M1.0            // 正转继电器
```

```
LRD
LD      I0.1            // 反转按钮
O       M1.1
ALD
AN      I0.0
=       M1.1            // 反转继电器
LPP
LPS
A       I0.3            // 高速点
MOVW    10, VW0         // 高速点预设时间
LPP
AN      I0.3            // 低速点
MOVW    100, VW0        // 低速点预设时间
Network 2               // 决定起始步位及循环计时程序段
LD      M1.0
O       M1.1
LPS
AN      T34
TON     T33, 10         // 步进间隔时间
LRD
A       T33
TON     T34, VW0        // 每一步所用的时间（步进速度）
LPP
AN      M0.0
AN      M0.1
AN      M0.2
AN      M0.3
AN      M0.4
AN      M0.5
AN      M0.6
AN      M0.7
MOVB    1, MB0          // 初始置位用于步进循环
Network 3               // 决定循环方向程序段
LD      T33
EU
```

```
LPS
A          M1.1
RRB        MB0, 1            // 右循环（步进电动机反向转动）
LPP
A          M1.0
RLB        MB0, 1            // 左循环（步进电动机正向转动）
Network 4                    // 决定 A 相得电与失电程序段
LD         M0.0              // 使 A 相得电
O          M0.1              // 使 A 相得电
O          M0.7              // 使 A 相得电
LD         M0.2              // 使 A 相失电
O          SM0.1             // 使 A 相失电
ON         I0.2              // 使 A 相失电
O          M0.6              // 使 A 相失电
NOT
LPS
A          Q0.0
=          Q0.0              //  A 相线圈
LPP
ALD
O          Q0.0
=          Q0.0              //  A 相线圈为一个 RS 触发器
Network 5                    // 决定 B 相得电与失电程序段
LD         M0.1              // 使 B 相得电
O          M0.2              // 使 B 相得电
O          M0.3              // 使 B 相得电
LD         M0.4              // 使 B 相失电
O          SM0.1             // 使 B 相失电
ON         I0.2              // 使 B 相失电
NOT
LPS
A          Q0.1
=          Q0.1              //  B 相线圈
LPP
ALD
```

```
O          Q0.1
=          Q0.1              //  B 相线圈为一个 RS 触发器
Network 6                    // 决定 C 相得电与失电程序段
LD         M0.3              // 使 C 相得电
O          M0.4              // 使 C 相得电
O          M0.5              // 使 C 相得电
LD         M0.6              // 使 C 相失电
O          SM0.1             // 使 C 相失电
ON         I0.2              // 使 C 相失电
NOT
LPS
A          Q0.2
=          Q0.2              //  C 相线圈
LPP
ALD
O          Q0.2
=          Q0.2              //  C 相线圈为一个 RS 触发器
Network 7                    // 决定 D 相得电与失电程序段
LD         M0.5              // 使 D 相得电
O          M0.6              // 使 D 相得电
O          M0.7              // 使 D 相得电
LD         M0.1              // 使 D 相失电
O          M0.4              // 使 D 相失电
ON         I0.2              // 使 D 相失电
O          SM0.1             // 使 D 相失电
NOT
LPS
A          Q0.3
=          Q0.3              //  D 相线圈
LPP
ALD
O          Q0.3
=          Q0.3              //  D 相线圈为一个 RS 触发器
Network 8                    // 停止时清零程序段
LD         I0.2              // 停止按钮
```

```
MOVB   0, MB0              // 停止时将决定循环位的字节清零
```

例十三、两台 PLC 主从式通信

1. 控制要求

这是两台 PLC 主从式通信的例子，通过这个例子我们应了解两台 PLC 间通信都应建立哪些初始化程序，主站怎样读取从站的数据又怎样将自己的数据写到从站中去，数据的通信是以变量寄存器为通道来实现的，这些寄存器不是唯一的，但只要建立了第一个后面的就要连续的相对应的使用（也就是说成组使用）。我们这个例子想达到的控制目的是在主站中用 I0.1 作为输入信号建立一个字节加一指令，送给从站的输出口显示出来，同时在主站中也累计变化过程，当数累加到 6 时，主站再给从站一个信号，从站接到这个信号后用自己的输入信号 I0.0 发给主站输出口点动信号。整个过程能说明只要建立好初始化关系，主站输入信号的逻辑关系能够控制从站的输出，反过来从站的输入信号也能控制主站的输出。像这个例子当中有个限制条件，就是只有当主站给从站的数累加到 6 以后，从站发给主站的信号才有效，在这之前主站是接不到从站信号的。

2. 程序设计

这种通信方式的主角就是主站，它让从站干什么，从站就干什么，同时它还可受控于从站，实质上就是数据读写。读写的区域范围由主站来定，哪些数据可以写给从站又有哪些数据找从站要，都是编程时需定好的，如本例中写给从站的数据是主站中 MB0 与 MB1 这两个字节，找从站要的数据是从站中 MB1 这一个字节。STEP 7-Micro/WIN 编程软件默认的单台 PLC 的地址是 2，现在是两台 PLC，如地址相同是不能通信的，怎么办？只好通过编程软件先把地址区分开，然后再分别给 PLC 下载各自的程序。按规定 PLC 的地址只能从 2 开始往后排，在本例中我们看到主站地址是 2，从站地址是 3，地址 2 好办，编程软件可以自己找到，地址 3 就要经过设置才能改变，如下介绍设置过程：打开编程软件如图 4-33 所示，用鼠标左点“检视”下面的系统块，显示界面如图 4-34 所示，在此看到端口 0 和端口 1 处的 PLC 地址都是 2，鼠标左点此口的上箭头，把 2 都变成 3，如图 4-35 所示，然后点“确认”，这时界面又回到图 4-33，点“▼”下载键把端口的设置下载给 PLC，然后用鼠标左点“检视”下面的通讯，通信结束后的界面如图 4-36 所示，发现这台 PLC 的地址已变成 3，点“确认”至此给 PLC 改地址的任务已完成，把相对应的程序送进去，在将两台 PLC 的模式开关都拨到“RUN”位置，就可以工作运行了。

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，确定主站与从站，配好两台 PLC 之间的通信电缆。主从式通信简单实惠，容易实现，难点与重点是主站的编程，读写区域与数据长度不能搞乱。控制系统 PLC 接线图如图 4-37 所示。

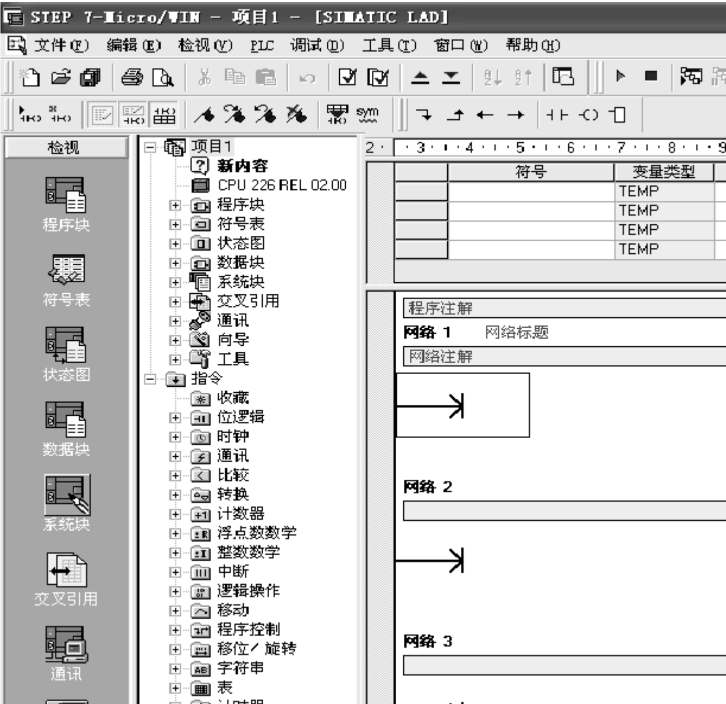


图 4-33　编程软件初始界面

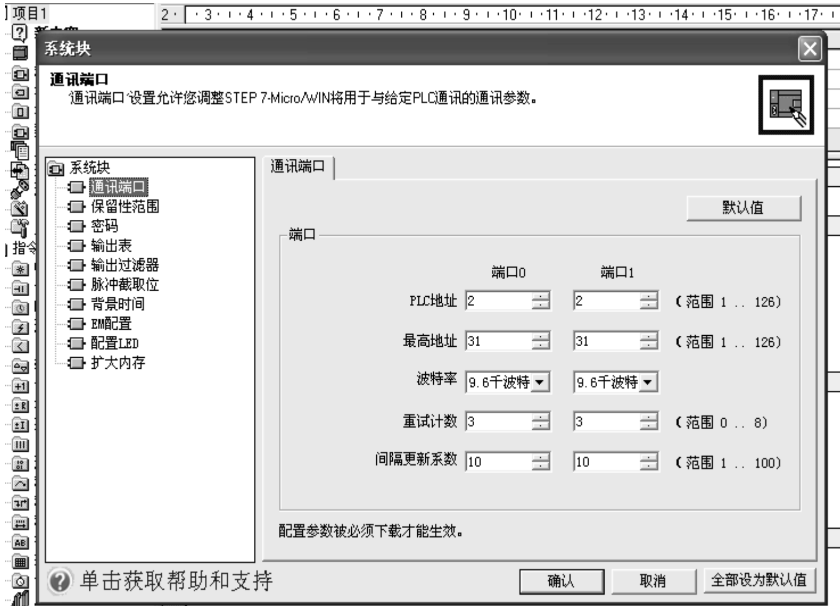


图 4-34　修改 PLC 地址的界面

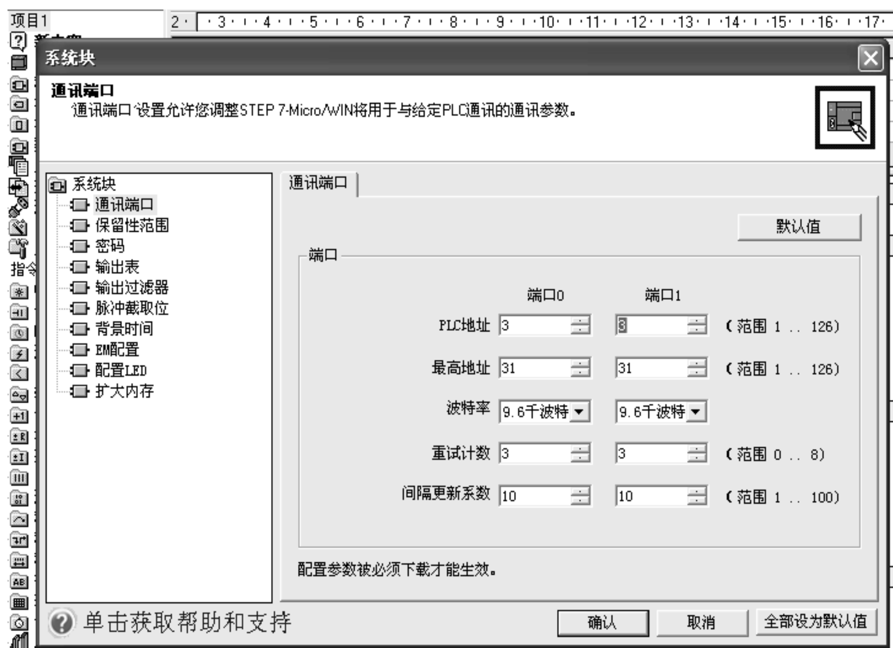


图 4-35　将地址 2 变成 3



图 4-36　通信后已搜到 PLC 的地址

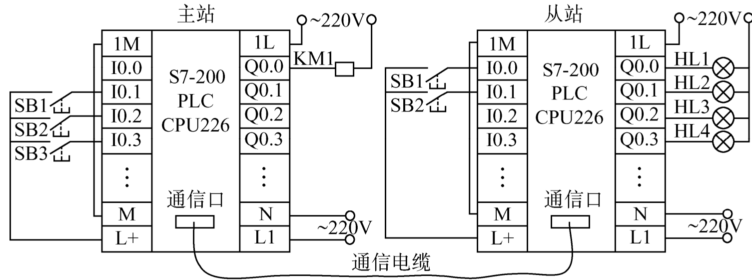


图 4-37 主从式通信控制系统 PLC 接线图

（2）控制程序梯形图如图 4-38 所示。

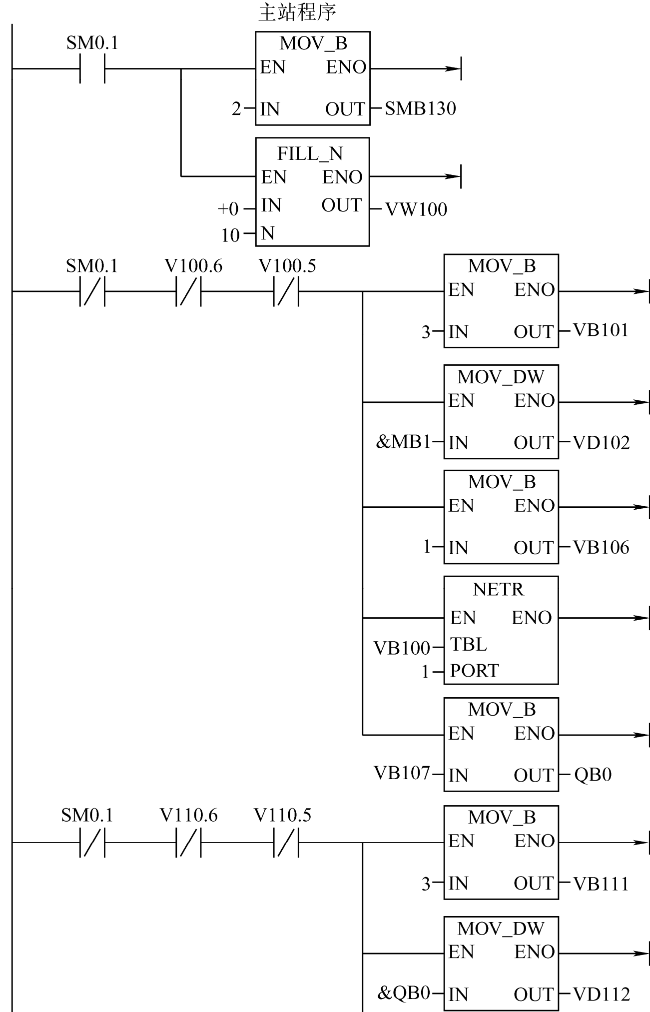


图 4-38 主从式通信控制程序梯形图

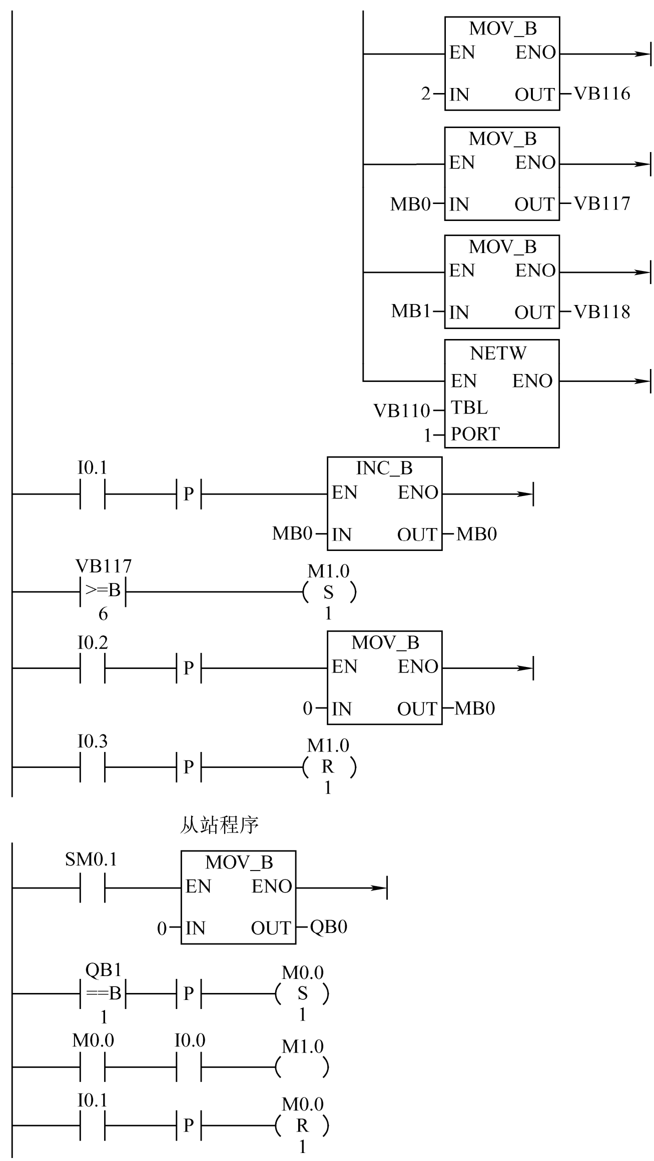


图 4-38　主从式通信控制程序梯形图（续）

（3）程序的语句表及注释如下：

主站程序

```
Network 1                  // 程序注释
LD      SM0.1
MOVB    2, SMB130          // 用通信端口 1 进行主从式通信
```

```
FILL    +0, VW100, 10    // 将 VB100 开始的 20 个变量寄存器都填充为 0
Network 2
LDN     SM0.1
AN      V100.6
AN      V100.5
MOVB    3, VB101         // 指定从站地址为 3
MOVD    &MB1, VD102      // 指向读取从站数据的位置
MOVB    1, VB106         // 读取数据的字节长度
NETR    VB100, 1         // 通过 1 号通信口由 VB100 开始的一列表完成读取
MOVB    VB107, QB0       // 读取进来的数据送到 QB0 处显示出来
Network 3
LDN     SM0.1
AN      V110.6
AN      V110.5
MOVB    3, VB111         // 指定从站地址为 3
MOVD    &QB0, VD112      // 指向送入从站数据的位置
MOVB    2, VB116         // 送出数据的字节长度
MOVB    MB0, VB117       // 将主站的 MB0 送给 VB117
MOVB    MB1, VB118       // 将主站的 MB1 送给 VB118
NETW    VB110, 1         // 通过 1 号通信口由 VB110 开始的一列表完成送出
Network 4
LD      I0.1             // 主站的字节加 1 输入信号
EU
INCB    MB0              // 字节 MB0 接到输入信号就自动加 1
Network 5
LDB>=   VB117, 6         // 当 VB117 中的数据大于等于 6 时
S       M1.0, 1          // 给 MB1 中的第一位置 1
Network 6
LD      I0.2             // 给 MB0 清零的输入信号
EU
MOVB    0, MB0           // 给 MB0 清 0
Network 7
LD      I0.3             // 给 M1.0 复位的输入信号
EU
R       M1.0, 1          // 给 M1.0 复位
```

从站程序

```
Network 1
LD      SM0.1
MOVB    0, QB1          // 给从站 QB1 清 0
Network 2
LDB=    QB1, 1          // 当 QB1 等于 1 时
EU
S       M0.0, 1         // 给 M0.0 置位
Network 3
LD      M0.0
A       I0.0
=       M1.0            // 当 M0.0 置位后再按 I0.0 按钮 M1.0 就会得电
Network 4
LD      I0.1            // 使 M0.0 复位的按钮
EU
R       M0.0, 1         // M0.0 复位
```

例十四、30/5 型桥式起重机小车运行的 PLC 改造程序

1. 改造思路

原控制系统以凸轮控制器实现大车、小车及副钩的运行控制，都属于不对称分挡切除绕线式电机转子上串接的电阻，且都分为 5 档，现以小车为例编写控制过程的 PLC 改造程序。改造后将以一个主令开关加两个按钮替代原有的凸轮控制器。

2. 控制要求

小车可实现正反向运行，都设有终端限位开关，每个方向上都可分为 5 挡切除电阻，从而实现调速。只要小车运行，抱闸都同时打开。运行到位后，应逐段接入所有电阻，以减小抱闸摩擦及接触器触点电流。

3. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，确定运行方向，主令开关是一个 3 点位开关，利用它可以定小车的零位、正向运行及反向运行。除此主令开关之外还需 2 个按钮，1 个用来增加电阻，1 个用来切除电阻。主令开关及按钮的示意图如图 4-39 所示，控制小车运行的 PLC 接

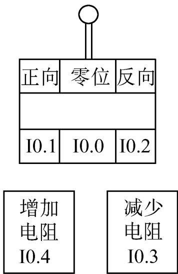


图 4-39　控制小车运行的主令开关及按钮示意图

线图如图 4-40 所示。

I/O分配:
I0.0——主令开关在零位(QS1)
I0.1——主令开关在正向位(QS1)
I0.2——主令开关在反向位(QS1)
I0.3——增挡位按钮(SB1)
I0.4——减挡位按钮(SB2)
I0.5——正向终端限位开关(SQ1)
I0.6——反向终端限位开关(SQ2)
Q0.1~Q0.5——对应5个挡位的切电阻接触器(KM1~KM5)
Q0.6——正向运行(KM6)
Q0.7——反向运行(KM7)

图 4-40　控制小车运行的 PLC 接线图

（2）控制程序梯形图如图 4-41 所示。

（3）程序的语句表及注释如下：

```
Network 1                     // 程序注释
LD      I0.0                  // 主令开关在零位
EU
O       SM0.1                 // 或者是在开始运行的第一个周期
MOVB    0, VB200              // 以上两方面都能使 VB200 清 0
Network 2
LDB=    VB200, 0
S       M0.0, 1               // 主令开关在零位时使 M0.0 置位
=       M0.2                  // 主令开关在零位时不允许 VB200 减 1
Network 3
LDB>=   VB200, 5
=       M0.1                  // VB200 的数据大于等于 5 时使 M0.1 闭合
Network 4
LD      M0.0
LPS
AN      M0.1
A       I0.3
EU
INCB    VB200                 // 按一次增挡位按钮 VB200 的数据就加一次 1
LPP
AN      M0.2
A       I0.4
```

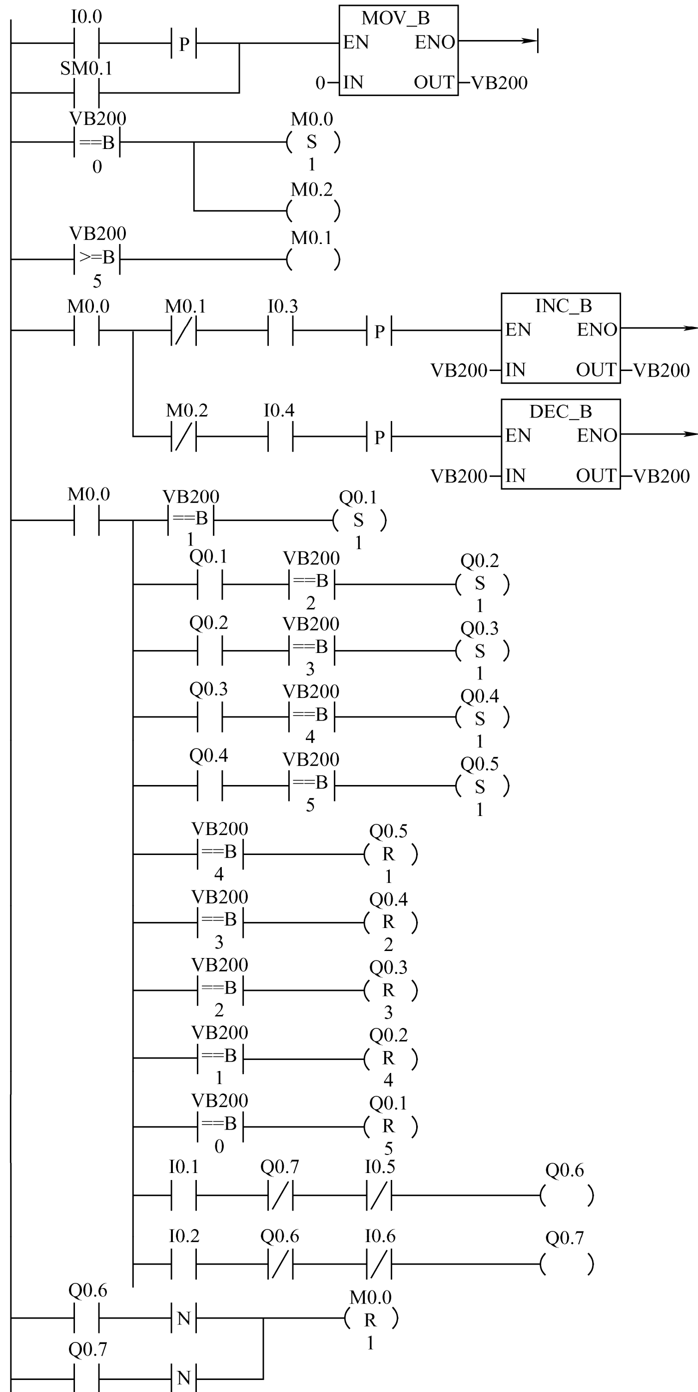


图 4-41　控制程序梯形图

```
EU
DECB    VB200          // 按一次减挡位按钮 VB200 的数据就减一次 1
Network 5
LD      M0.0
LPS
AB=     VB200, 1
S       Q0.1, 1        // 当 VB200 等于 1 时小车第一次切电阻加速
LRD
A       Q0.1
AB=     VB200, 2
S       Q0.2, 1        // 当 VB200 等于 2 时小车第二次切电阻加速
LRD
A       Q0.2
AB=     VB200, 3
S       Q0.3, 1        // 当 VB200 等于 3 时小车第三次切电阻加速
LRD
A       Q0.3
AB=     VB200, 4
S       Q0.4, 1        // 当 VB200 等于 4 时小车第四次切电阻加速
LRD
A       Q0.4
AB=     VB200, 5
S       Q0.5, 1        // 当 VB200 等于 5 时小车第五次切电阻加速
LRD
AB=     VB200, 4
R       Q0.5, 1        // 当 VB200 等于 4 时小车第一次加电阻减速
LRD
AB=     VB200, 3
R       Q0.4, 2        // 当 VB200 等于 3 时小车第二次加电阻减速
LRD
AB=     VB200, 2
R       Q0.3, 3        // 当 VB200 等于 2 时小车第三次加电阻减速
LRD
AB=     VB200, 1
R       Q0.2, 4        // 当 VB200 等于 1 时小车第四次加电阻减速
```

```
LRD
AB=     VB200, 0
R       Q0.1, 5         // 当 VB200 等于 0 时小车第五次加电阻减速
LRD
A       I0.1
AN      Q0.7
AN      I0.5            // 小车正向限位
=       Q0.6            // 小车正向运行
LPP
A       I0.2
AN      Q0.6
AN      I0.6            // 小车反向限位
=       Q0.7            // 小车反向运行
Network 6
LD      Q0.6
ED
LD      Q0.7
ED
OLD
R       M0.0, 1         // 保证小车从零速开始运行
```

例十五、外部输入信号中断

1. 控制要求

在 I0.1 的上升沿通过中断使 Q0.2 立即置位，在 I0.2 的下降沿通过中断使 Q0.2 立即复位。

2. 程序分析

SM0.1 是一个特殊继电器,它的作用是当 PLC 由 STOP 转为 RUN 时，SM0.1 只在第一个扫描周期为 ON 状态，从第二个扫描周期开始 SM0.1 就始终处在 OFF 状态。

在下面的程序中看不到对应的输入点，这是因为使用了中断功能，只要我们把程序设计好，当符合某个中断事件的信号出现时，程序就会执行中断事件指向的中断程序。图 4-42 中的 SB1 闭合时的上升沿可产生 2 号中断事件，SB2 断开时的下降沿可产生 5 号中断事件，KM1 将对应这两个输入信号有一个吸合过程。

3. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配。对应信号关系的

PLC 接线图如图 4-42 所示。

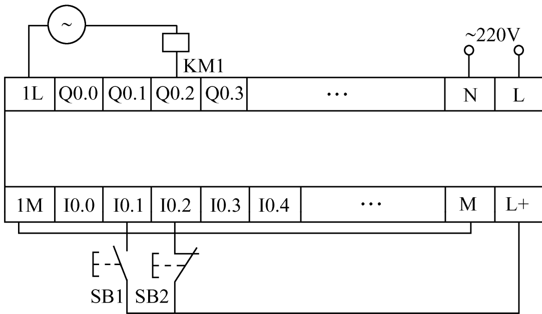


图 4-42 信号关系 PLC 接线图

（2）控制系统梯形图程序如图 4-43 所示，在 STEP 7 编程软件中主程序与中断程序要分别编写，各自都有自己的窗口，操作方法参见第六章。

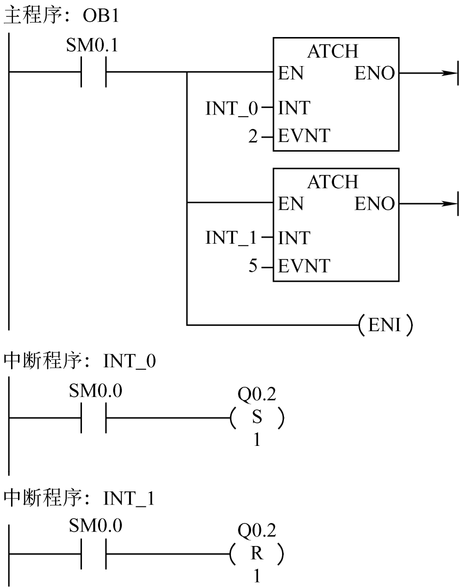


图 4-43 控制程序梯形图

（3）程序的语句表及注释如下：

主程序：OB1

```
Network 1                    // 程序注释
LD       SM0.1               // 主程序初始化
ATCH     INT0, 2             // 当发生 2 号中断事件时执行 0 号中断程序
ATCH     INT1, 5             // 当发生 5 号中断事件时执行 1 号中断程序
```

```
ENI                          // 开通中断
中断程序：INT_0
Network 1
LD      SM0.0
S       Q0.2, 1              // 执行 0 号中断程序时，使 Q0.2 置位
中断程序：INT_1
Network 1
LD      SM0.0
R       Q0.2, 1              // 执行 1 号中断程序时，使 Q0.2 复位
```

例十六、利用“定时中断”的彩灯循环左移

1. 控制要求

利用“定时中断”给 8 位彩灯循环左移。先设定 8 位彩灯在 QB0 处显示，并设初始值“9”，然后每隔 1s 彩灯循环左移一位。控制按钮（SB1）选 I0.1 按一次开始，再按一次停止，停止后彩灯全灭。本题的特点是利用特殊继电器 SMB34 的定时产生第十号中断事件去执行 0 号中断程序。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配。对应信号关系的 PLC 接线图如图 4-44 所示。

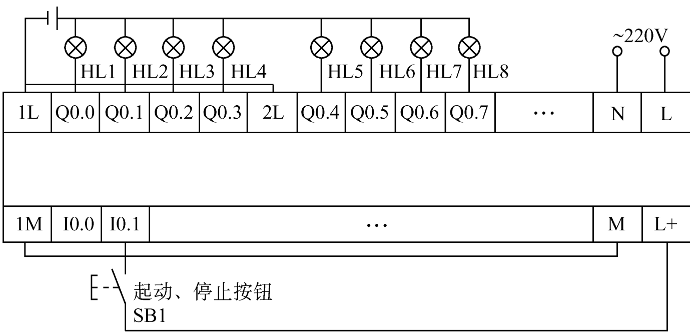


图 4-44　彩灯循环 PLC 控制接线图

（2）控制系统梯形图程序如图 4-45 所示，在 STEP 7 编程软件中主程序与中断程序要分别编写，各自都有自己的窗口，操作方法参见第六章。

（3）程序的语句表及注释如下：

```
主程序：OB1
Network 1                    // 程序注释
LD      I0.1                 // 按奇数次时起动，按偶数次时停止
```

```
EU
LD      I0.1
EU
A       M0.0
NOT
```

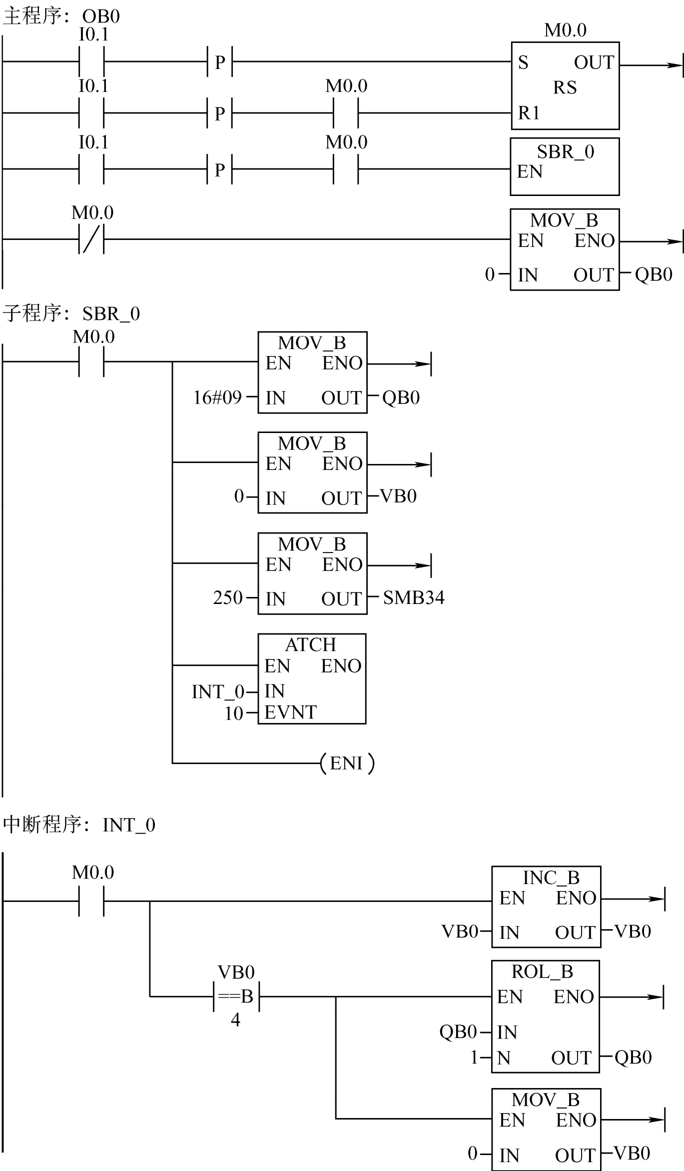


图 4-45　控制程序梯形图

```
LPS
A       M0.0
=       M0.0
LPP
ALD
O       M0.0
=       M0.0              // 决定运行或停止
Network 2
LD      I0.1
EU
A       M0.0
CALL    SBR0
Network 3
LDN     M0.0
MOVB    0, QB0            // 在停止状态下给 QB0 清 0
子程序：SBR_0
Network 1
LD      M0.0
MOVB    16#09, QB0        // 进入子程序给 QB0 置初始值 9
MOVB    0, VB0            // 初始化时给 VB0 清 0
MOVB    250, SMB34        // 初始化时给特殊继电器 SMB34 设定中断时间
ATCH    INT0, 10          // 当第 10 号中断事件产生时执行 0 号中断程序
ENI                       // 开通中断
中断程序：INT_0
Network 1
LD      M0.0

INCB    VB0               // 执行一次中断 VB0 加一次 1
AB=     VB0, 4            // 执行 4 次中断相当于 1s
RLB     QB0, 1            // 隔 1s 彩灯循环左移一位
MOVB    0, VB0            // 给变量寄存器清 0，再去等 4 次
```

例十七、Z3050 型摇臂钻床的 PLC 改造

1. 控制要求

主轴电动机 1M 随时都可以起停，并保持。起动按钮是 SB2，停止按钮是 SB1，

接触器是 KM1，热继电器是 FR1，摇臂的升降控制：SB3 是摇臂上升按钮，SB4 是下降按钮，SQ1U 是上升终端限位开关，SQ1D 是下降终端限位开关，KM2 是上升接触器，KM3 是下降接触器。假设想使摇臂上升，就要按 SB3 按钮，这时如果摇臂是处在抱住立柱的位置，那么 SQ2 限位开关的常开点是断开的，常闭点就是闭合的了，这样控制油泵放松的接触器 KM4 与电磁铁 YA 就先得电，使摇臂与立柱松开，当放松到位时，SQ2 动作，常开触点闭合，常闭触点断开，这样摇臂就可以上升了。下降也是同样的动作过程。当上升结束时，松开 SB3 按钮，KT、KM2、KM3、KM4 全部失电，经过 KT 延时闭合的常闭触点的延时后，油泵夹紧方向的接触器 KM5 得电吸合。同时 YA 继续得电吸合直到夹紧到位，SQ3 限位开关动作，KM5 与 YA 全部失电。立柱与主轴箱的夹紧与放松：SB5 是立柱放松按钮，SB6 是立柱夹紧按钮。本次改造是以 1994 年中国劳动出版社出版，劳动部培训司组织编写的《维修电工生产实习》（第 2 版）为脚本。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配。机床改造的基本思想是遵循原有工作原理，只改造控制部分，动力线路不变，辅助的照明灯、指示灯不变。Z3050 型摇臂钻床的电气原理图如图 4-46 所示，PLC 对外接线 I/O 分配图如图 4-47 所示。

（2）控制系统梯形图程序如图 4-48 所示。

（3）程序的语句表及注释如下：

```
                              程序注释
Network 1
LD     I0.0                   // 主轴起动
O      Q0.0
A      I0.1                   // 主轴停止
A      I1.2                   // 主轴电动机过载保护
=      Q0.0                   // 主轴电动机接触器线圈
Network 2
LD     I1.3                   // 液压泵电动机过载保护
LPS
AN     I1.0                   // 摇臂放松到位此点闭合摇臂才可上升或下降
LPS
A      I0.2                   // 摇臂上升控制按钮
A      I0.6                   // 摇臂上升到位此点断开
AN     I0.3                   // 如按摇臂下降按钮上升就会停止
AN     Q0.2                   // 如摇臂正在下降就不会有上升
```

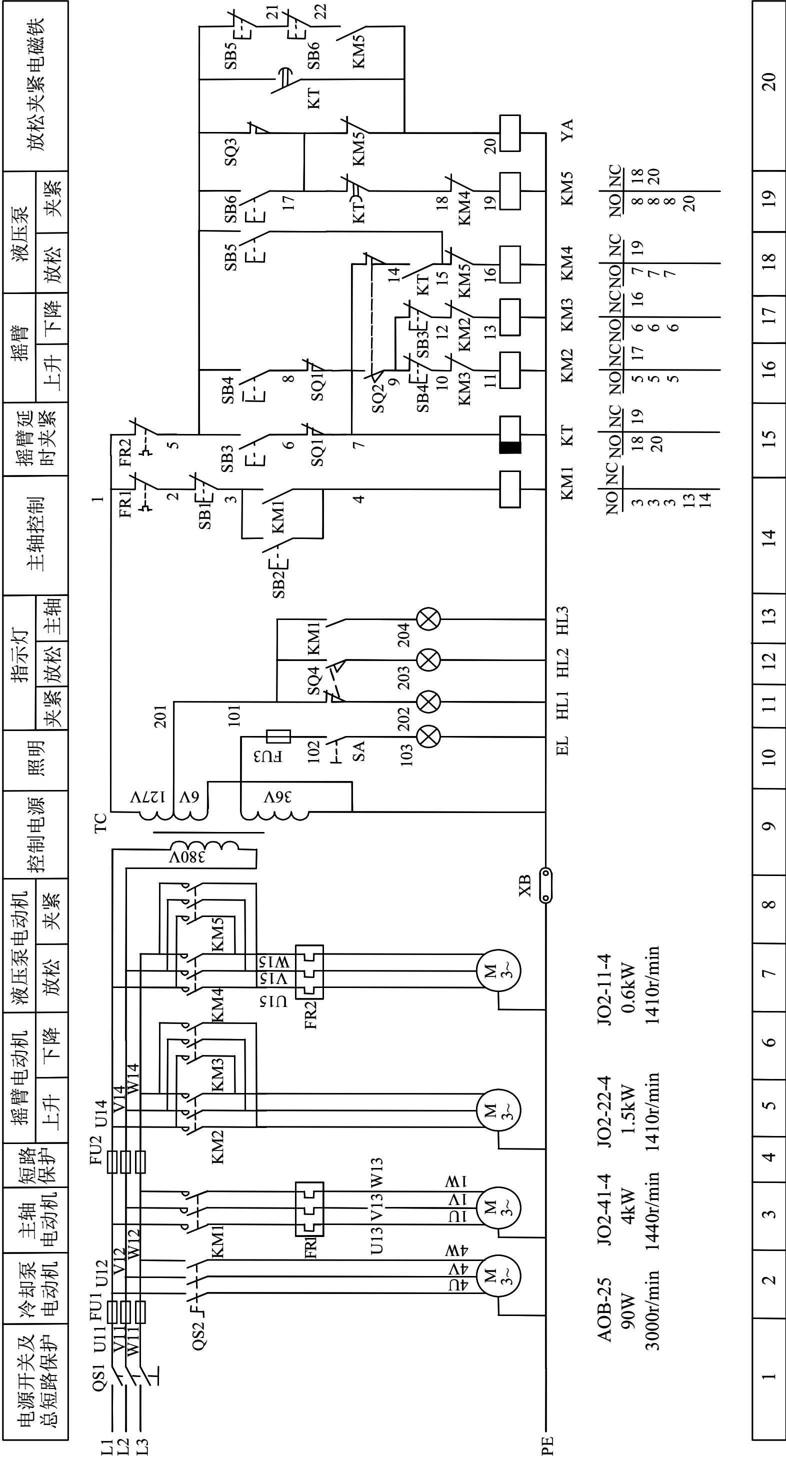


图 4-46 Z3050 型摇臂钻床电气原理图

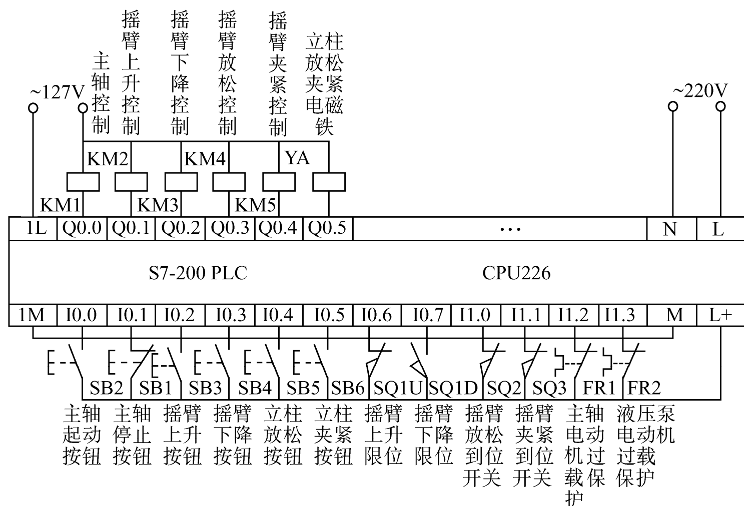


图 4-47 PLC 对外接线 I/O 分配图

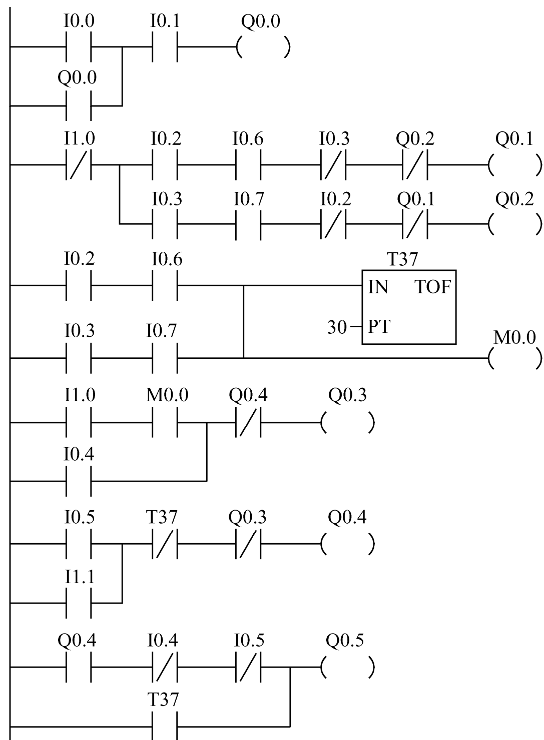


图 4-48 摇臂钻床控制程序梯形图

```
=          Q0.1              // 摇臂电动机上升接触器线圈
LPP
```

```
A      I0.3          // 摇臂下降控制按钮
AN     I0.7          // 摇臂下降到位此点断开
AN     I0.2          // 如按摇臂上升按钮下降就会停止
AN     Q0.1          // 如摇臂正在上升就不会有下降
=      Q0.2          // 摇臂电动机下降接触器线圈
LRD
LD     I0.2
A      I0.6
LD     I0.3
AN     I0.7
OLD
ALD
TOF    T37, 30       // 无论上升或下降都要使断电延时继电器得电
=      M0.0          // 无论摇臂上升或下降都应使摇臂先放松
LRD
LD     I1.0          // 摇臂放松到位此点断开
A      M0.0
O      I0.4          // 立柱和主轴箱松开按钮
ALD
AN     Q0.4          // 夹紧与松开的互锁
=      Q0.3          // 立柱和主轴箱松开接触器线圈
LRD
LD     I0.5          // 立柱和主轴箱夹紧按钮
O      I1.1          // 摇臂夹紧到位此点断开
ALD
AN     T37           // 摇臂上升或下降后要延时一段时间再夹紧
AN     Q0.3          // 松开与夹紧的互锁
=      Q0.4          // 立柱和主轴箱夹紧接触器线圈
LPP
LD     Q0.4          // 在摇臂夹紧过程中电磁铁也随之动作
AN     I0.4
AN     I0.5
O      T37
LD     I0.5
O      I1.1
```

```
AN      Q0.4
OLD
ALD
=       Q0.5          // 与摇臂升降同时动作的电磁铁的控制线圈
```

例十八、用 TD200 文本显示器监控密码锁开启

1. TD200 文本显示器简介

TD200（Text Display 200）是专用于 S7-200 系列的文本显示和操作员界面，它支持中文操作和中文显示。

TD200 有许多功能，现就与本题有关的内容简介如下，如果通过这个例子我们知道 TD200 如何使用了，那接下来还可以开发它的更多功能。

文本显示是指 TD200 上的显示窗口可以显示文本，为什么称为文本不称为文字？是因为它既可以显示中文还可以显示外文字母与数据，具体显示什么要由我们设置的，而且数据是可变的，也就是说能反应即时值，比如可以让它告诉我们小车往返几周了、高速计数器记录多少数码了、桥式起重机哪个环节正在工作及接切电阻数目等。文本显示每次可显示两行，每行 20 个字符。这两行每次可显示一条信息也可显示两条信息，具体怎么显示还是要由我们设置。

操作员界面是指 TD200 上除了显示窗口外还有 9 个按键可以操作，其中的▲、▼、ESC、ENTER 这四个键是决定 TD200 本身设置的，通过它们可以设定 TD200 地址、通信波特率、CPU 地址、参数块地址等。这些功能本题都不用设置，在这里就不做介绍了。另外 5 个键是 F1～F4 及 SHIFT 键，因本题要利用这 5 个键，所以下面介绍一下，通过设置可将这 5 个键形成 8 个按钮，也就是 F1～F4 可当做 4 个按钮，SHIFT 键与 F1～F4 其中之一组合又可以形成 4 个按钮，这样就是 8 个按钮了。本题并没有将 8 个按钮都用上，这些按钮相当于是 S7-200 输入信号的扩展，通过设置就可当做 S7-200 PLC 的输入信号使用了。

TD200 包装盒中提供了专用电缆（TD/CPU 电缆），用此电缆将 TD200 与 S7-200 CPU 连接起来。电缆能从 CPU 通讯口上取得 TD200 所需的 24V 直流电源。1 个 S7-200 CPU 的通信口最多可以连接 3 个 TD200，这 3 个 TD200 所访问的参数块可以相同也可不同，而 1 个 TD200 只能与 1 个 S7-200 CPU 建立连接。TD200 包装盒中的器件如图 4-49 所示。各种器件的作用见表 4-2。

STEP 7Micro/WIN 编程软件提供了集成的 TD200 组态工具，TD200 的组态信息全部保存在 S7-200 CPU 中，可以方便地更换 TD200 而不必重新组态。

2. 控制要求

有一个密码锁，它有 6 个按键 SB1～SB6，其控制要求为

① SB1 为起动键，按下 SB1 键，才可进行开锁作业；

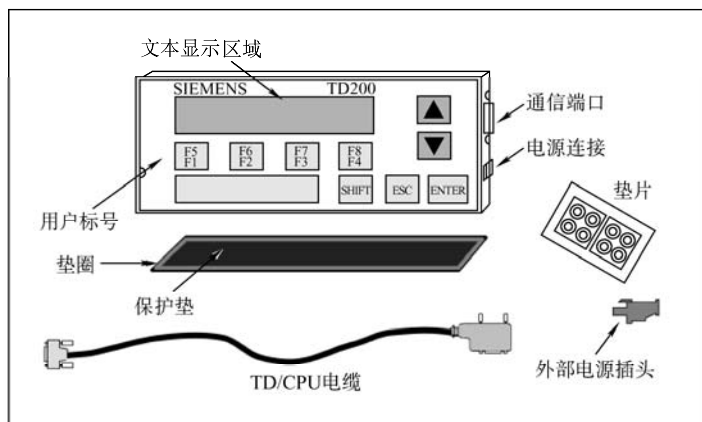


图 4-49　TD200 包装盒中的器件

表 4-2　包装盒内各种器件作用说明

部件	说　　明
文本显示区	文本显示区为一个背光液晶显示（LCD），可显示两行信息，每行 20 个字符，可以显示从 S7-200 接收来的信息
垫圈	TD200 随机提供一个垫圈，用于在恶劣环境安装时使用
通信端口	通信端口是一个 9 针 D 形连接器，它使你可以用 TD/CPU 电缆把 DT200 连接到 S7-200 CPU
电源连线	可以通过 TD200 右边的电源接入口，把外部电源连接到 TD200，当使用 TD/CPU 电缆时，则不需要外部电源
TD/CPU 电缆	通过 TD/CPU 电缆可以与 TD200 通信并向其提供电源，它是 9 针直通的电缆，与 TD200 随机提供
用户标签	用户标签是一个插入式标签，可以根据你的应用改制功能键标签
键	TD200 有 9 个键，其中 5 个键提供预定义、上下文有关的功能，其于 4 个键用户可以提供其功能
垫片	包括有自粘的垫片，用于把 TD200 安装在安装面上

② SB2、SB5 为可按压键。开锁条件为：SB2 按压 3 次，SB5 按压 4 次。如果按上述规定按压，则 5s 后,密码锁自动打开。（SB2 与 SB5 没有先后顺序，先按谁都可以）；

③ SB3、SB4 为不可按压键，一旦按压，报警器就警报；

④ SB6 为停止键，按下 SB6 键，停止开锁作业。

3. 连接与设置

TD200 的外形见图 4-49，在 S7-200 PLC 通电前就应把 TD200 与 S7-200 PLC 连接好，可以用 TD/CPU 电缆方便地将两者通过通信端口连接起来，然后给 S7-200 通上电源，两者就都有电了。TD200 的显示窗口点亮并显示本 TD200 的型号与版

本，接下来又显示 CPU 的状态，这时可以通过计算机（已经通过通信口与 S7-200 PLC 建立好通信关系）上的 STEP 7Micro/WIN 编程软件进行设置（组态）。

设置（组态）是通过指令树中的向导来完成的，打开向导会出现如图 4-50 所示的画面，这是第一个画面，只是将向导的功能简介一下，直接单击“下一步”按钮即可。

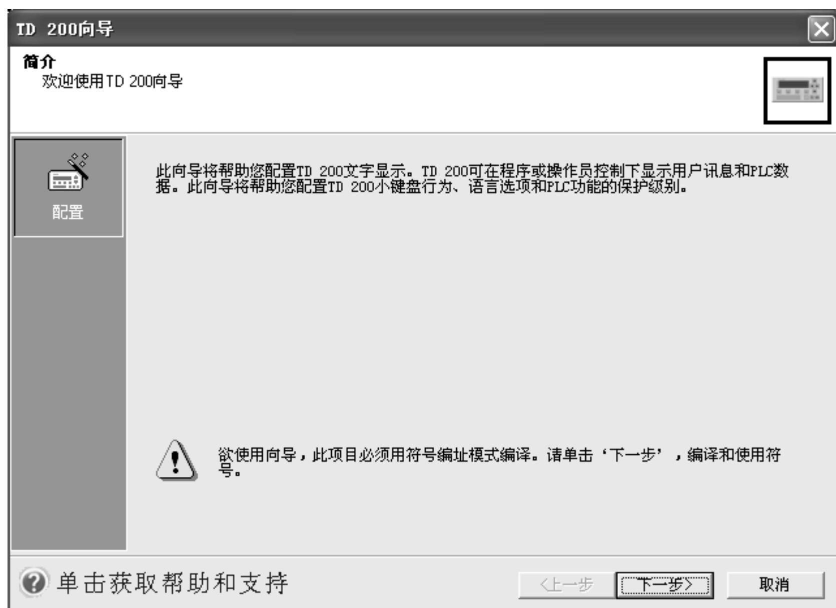


图 4-50　TD200 向导的简介

下一步的画面如图 4-51 所示，让我们选择 TD200 的版本，而 TD200 的版本也是非常容易查到的，一是给 TD200 上电在初始化画面上会显示出 TD 的型号和版本；另一种方法是在 TD 的背面找到其型号和版本号。本题使用的是 3.0 版，所以就在 TD200 3.0 版那里单击即可，然后再单击“下一步”按钮。

下一步的画面如图 4-52 所示，这一步与本题设置无关，什么也不用动，直接单击“下一步”按钮即可。

下一步的画面如图 4-53 所示，这一步是选择语言，选择中文并选简体中文，然后单击“下一步”按钮即可。

下一步的画面如图 4-54 所示，这一步是选择按钮的功能，在画面中的“按钮动作”栏下的白色框内用鼠标单击会出现▼按钮，单击此按钮会出现两个可选项，一是“设置位”，另一个是“瞬时接触”，因本题需要此键当按钮使用所以就选瞬时接触，本题使用 6 个按钮，我们就把前 6 个都选为瞬时接触，然后点“下一步”按钮进入下一个画面。



图 4-51　选择 TD200 型号和版本

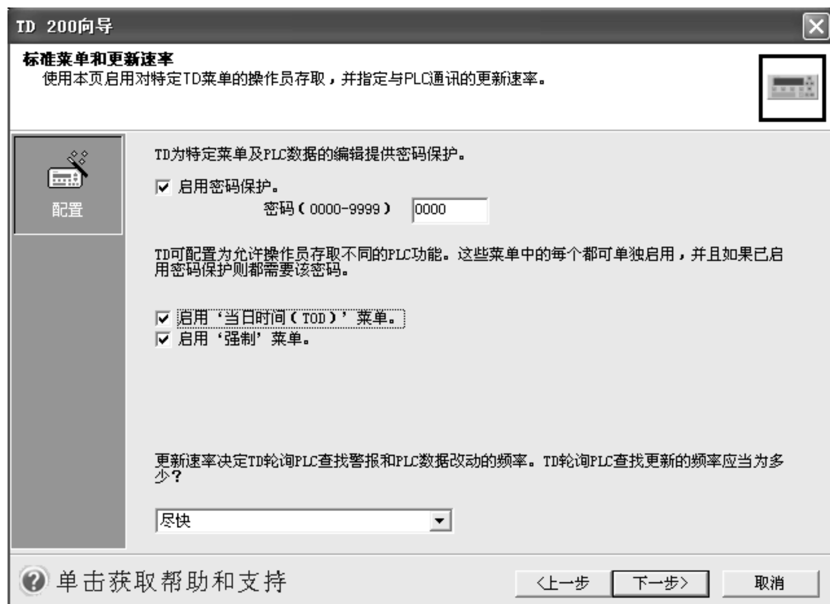


图 4-52　定义 TD 功能及数据更新速率

下一步的画面如图 4-55 所示，这一步显示 TD 配置已完成，实际上设置远没有结束，只是将 TD 上的那几个按键设置完成，需要 TD 显示的各项内容还没设置。

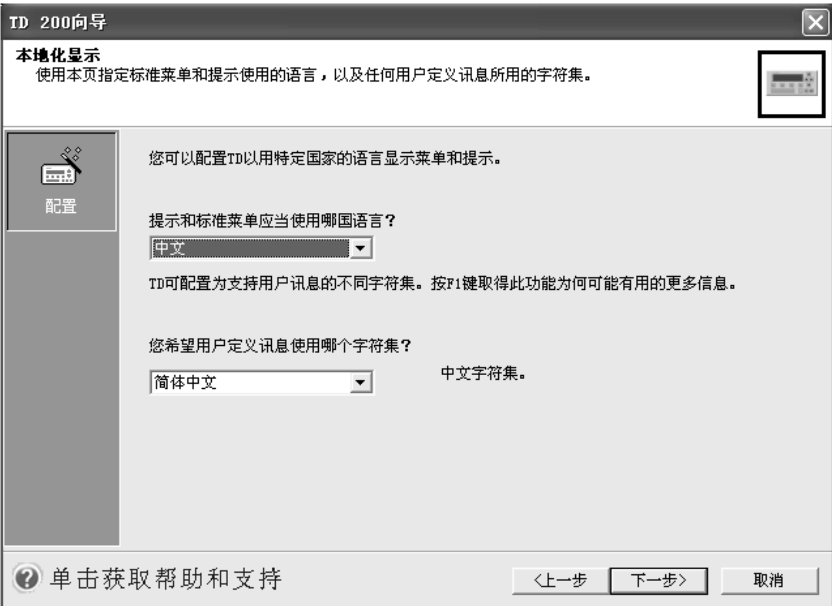


图 4-53　选择系统菜单语言及用户菜单语言

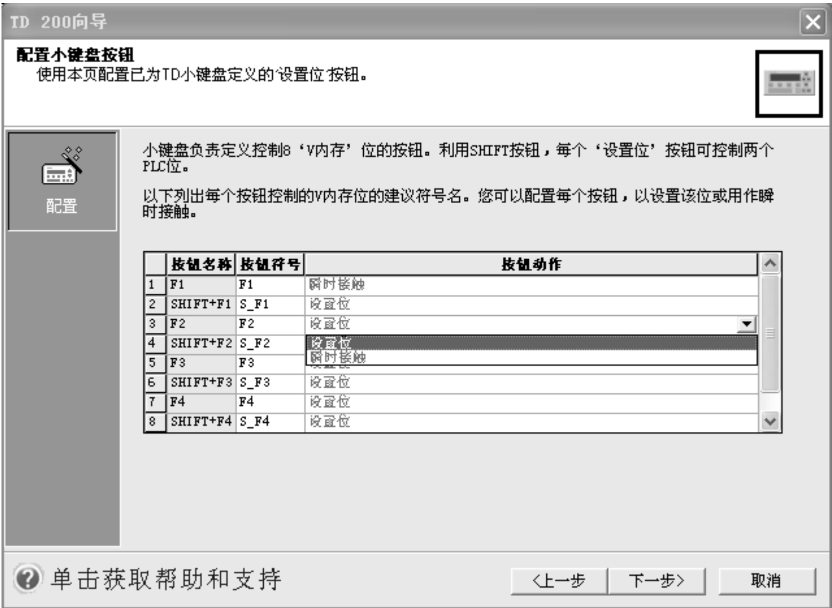


图 4-54　设置按键工作方式

这时单击画面中的“警报”，然后再单击“下一步”按钮，画面会变成如图 4-56 所示。这时候的画面问需要显示的信息是每次一行还是两行，如选择每次一行的

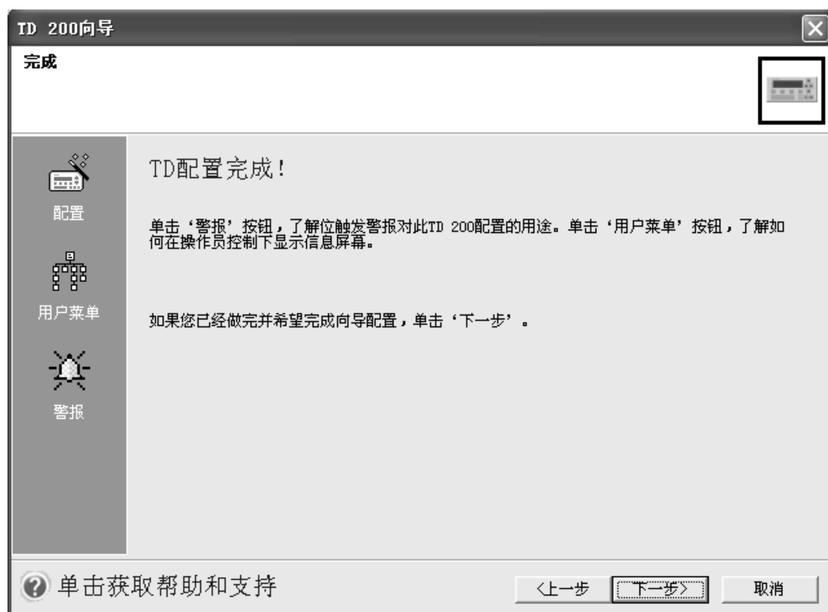


图 4-55　选择警报设置

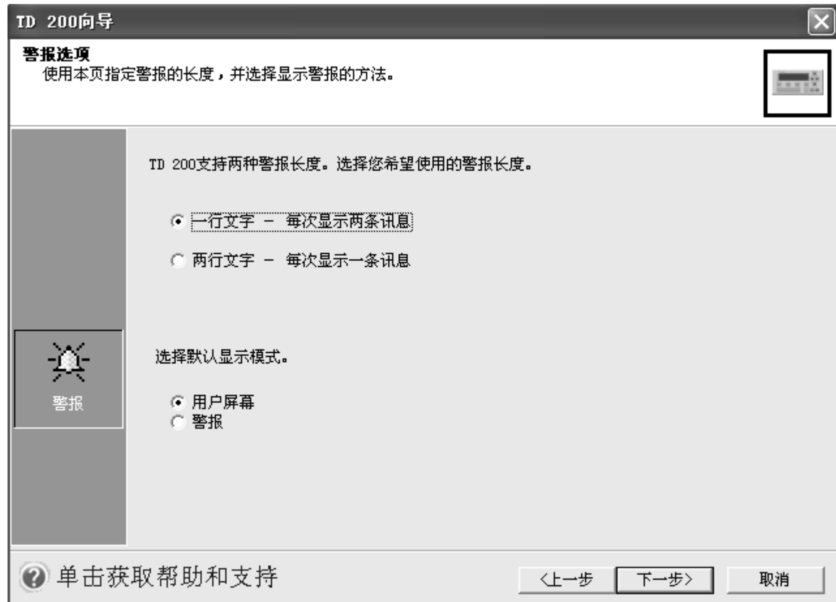


图 4-56　选择文本显示条数

每次可以显示两条信息，而本题某个环节需要同一时刻显示两条信息，所以我们就选择“一行文字-每次显示两条讯息”的那种，圆形空白处单击即可，在选择显

示模式处单击“警报”，然后单击“下一步”，弹出一个对话框问“您希望为此配置增加一则警报吗？”单击“是”按钮，然后画面出现了如图 4-57 所示的内容，在此时的设置是本题的重要环节。

将图 4-57 中的有关功能介绍如下：

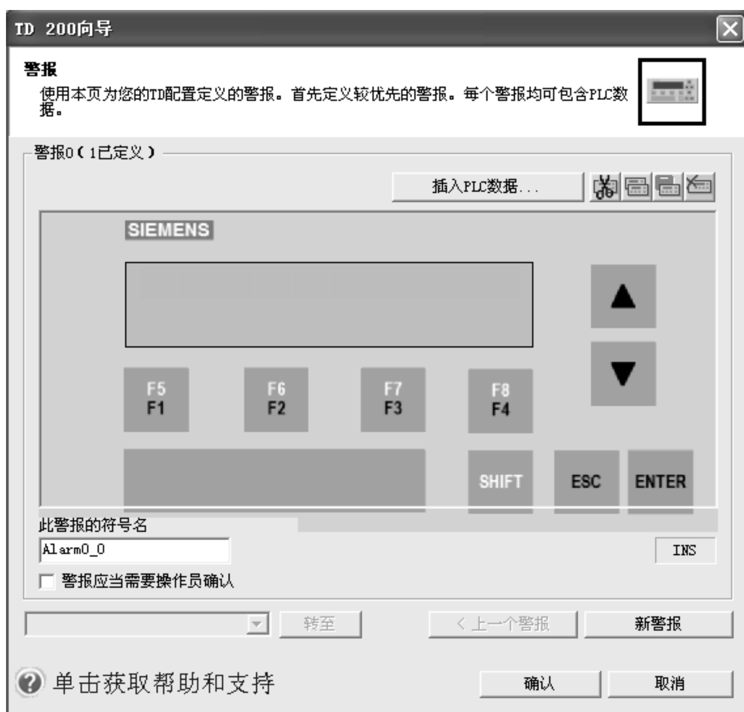


图 4-57　编辑显示文本信息

在左上角处有“警报 0（1 已定义）”几个字，这是说在这个画面中你输入的文本信息是由 0 号警报负责显示，假设这时在 SIEMENS 标示下面的矩形方框内输入“开锁工作开始”，那么一定要记住这条信息是由 0 号警报负责显示的，因为在后面还要为 0 号警报找到相对应的变量寄存器（也就是 V 寄存器），在编程时让这个 V 寄存器什么时候得电，这个显示窗口就什么时候显示这条信息，以此类推，后面的设置也是这样的。接下来往下看，在左下角有一框内写有“Alarm0_0”，这是此信息的符号名，它与 V 存储器也是一一对应的。这时如想输入第二条信息，那就用鼠标单击“新警报”按钮，出现的画面与图 4-57 是一样的，还是在显示窗口输入信息即可。后面还有第三条、第四条等依此类推，方法都是与第一条相同。需要显示的信息都输入完之后按“确认”按钮，所有的信息都会自动找到相对应的 V 存储器，后面还要介绍到哪里去找它们之间的对应关系表。接下来还有一个重点，就是如何使显示窗显示数据变量（即时值），这时输入如图 4-58 所示的文

本，然后单击“插入 PLC 数据……”按钮，弹出的画面如图 4-59 所示，问我们需显示的数据变量准备放到哪个 V 寄存器中，它给出的是 VW0，如不想改就记住此地址，如想改可在此处输入新的地址，后面的格式、小数点后位数等根据自己的需要设定。这个地址一定要记住，我们在编程时把需显示的可变数据送到相应

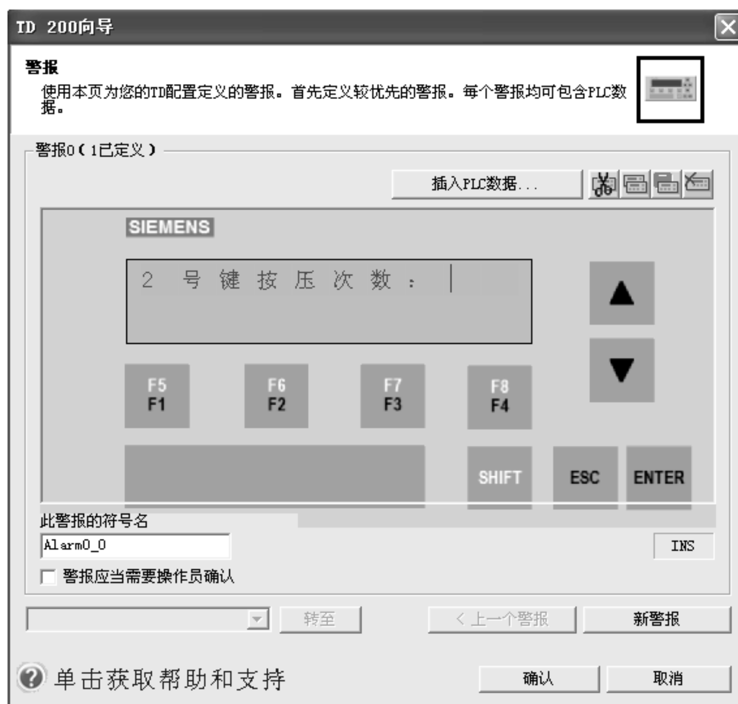


图 4-58　编辑显示可变文本信息



图 4-59　定义数据变量存放地址及格式

的 V 寄存器里，显示时就会看见此变量。单击“确认”按钮出现如图 4-60 所示的画面，与图 4-58 相比会发现在冒号的后面多了两个方框，等到运行起来需显示时，那个地方就会显示可变的数据了。

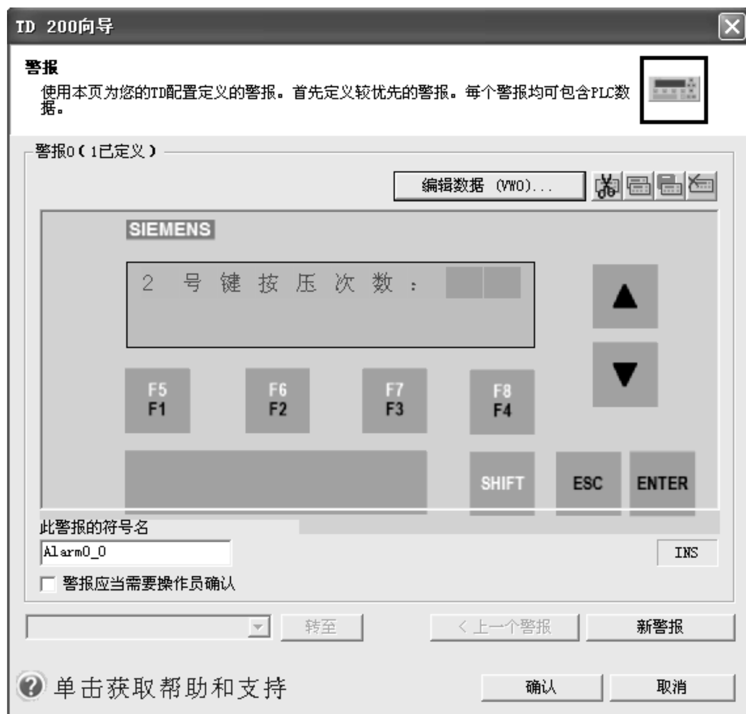


图 4-60 可显示数据变量的显示模式

当我们需显示的内容全部设置完成后，单击图 4-56 中的“确认”按钮，会弹出如图 4-61 所示的画面，单击“下一步”按钮会出现如图 4-62 所示的画面，这一画面是让我们定 TD 上的所有显示内容及按钮所对应的变量寄存器地址，我们应找未被占用的区域，定好后单击“下一步”按钮，所有的设置就全部完成了。

接下来是从哪里找到与 TD 相对应的变量寄存器？找到后把它们的一一对应关系记下来，剩下的工作就是编程了。在 STEP 7-Micro/WIN V4.0 版编程软件的指令树中单击符号表，这时会出来 3 个选项，我们单击“向导”按钮后又出现 TD_SYM_100，继续单击它会出现如图 4-63 所示的画面。在这里能找到我们所设置的按钮及显示信息对应的变量寄存器的地址，对照着它进行编程所设置的功能就会准确无误地都能实现。再接下来就要编写本题的控制程序了，在那里能够体现我们所设置的功能。还有一点需说明，本题的 TD 与变量寄存器的对应地址首地址选择的是 VB100，这样选择后的对应关系就如图 4-63 所示了。

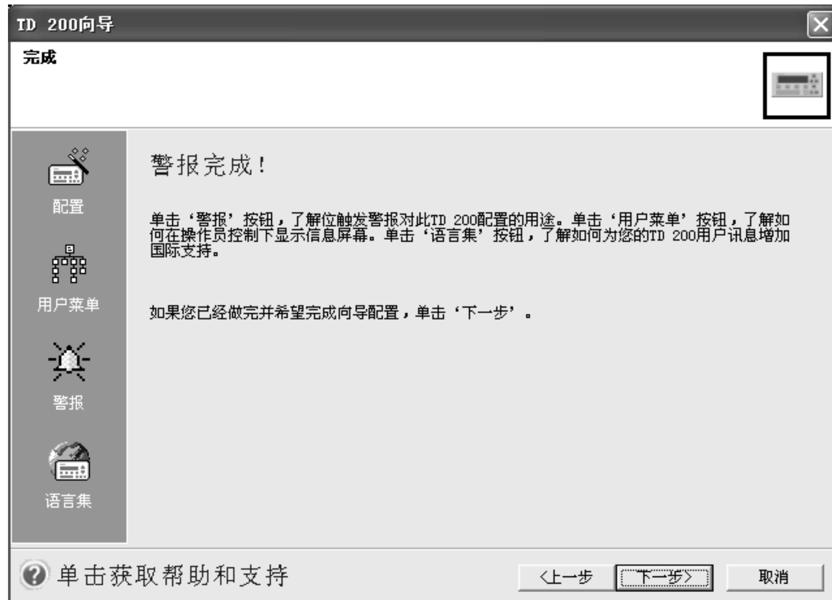


图 4-61　文本显示内容设置完毕

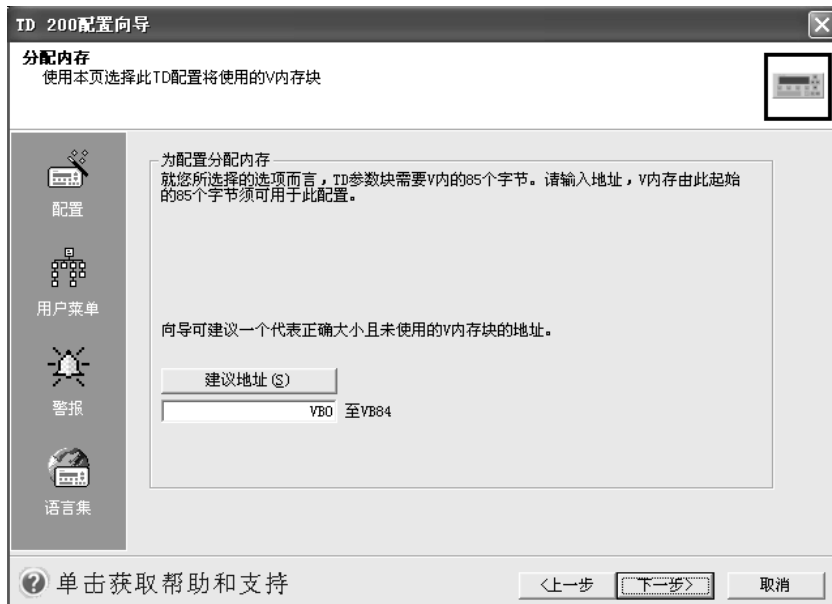


图 4-62　给 TD 定存储器地址

符号表

	符号	地址	注解
1	S_F4	V159.7	键盘按键"SHIFT+F4"已按下标志（瞬动触点）
2	F4	V157.3	键盘按键"F4"已按下标志（瞬动触点）
3	S_F3	V159.6	键盘按键"SHIFT+F3"已按下标志（瞬动触点）
4	F3	V157.2	键盘按键"F3"已按下标志（瞬动触点）
5	S_F2	V159.5	键盘按键"SHIFT+F2"已按下标志（瞬动触点）
6	F2	V157.1	键盘按键"F2"已按下标志（瞬动触点）
7	S_F1	V159.4	键盘按键"SHIFT+F1"已按下标志（瞬动触点）
8	F1	V157.0	键盘按键"F1"已按下标志（瞬动触点）
9	TD_CurScreen_100	VB163	TD 200 显示的当前屏幕（其配置起始于 VB100）。如无屏幕显示则设置为 16#FF。
10	TD_Left_Arrow_Key_100	V156.4	'左箭头'键按下时置位
11	TD_Right_Arrow_Key_100	V156.3	'右箭头'键按下时置位
12	TD_Enter_100	V156.2	'ENTER'键按下时置位
13	TD_Down_Arrow_Key_100	V156.1	'下箭头'键按下时置位
14	TD_Up_Arrow_Key_100	V156.0	'上箭头'键按下时置位
15	TD_Reset_100	V145.0	此位置位会使 TD 200 从 VB100 重读其配置信息。
16	Alarm0_5	V146.2	报警使能位 5
17	Alarm0_4	V146.3	报警使能位 4
18	Alarm0_3	V146.4	报警使能位 3
19	Alarm0_2	V146.5	报警使能位 2
20	Alarm0_1	V146.6	报警使能位 1
21	Alarm0_0	V146.7	报警使能位 0

用户定义1　POU 符号　TD_SYM_100

图 4-63　与 TD 相对应的变量寄存器的地址

4. 程序设计

（1）S7-200 PLC 与 TD200 文本显示器之间通信电缆连接示意图及 PLC 对外接线图如图 4-64 所示。

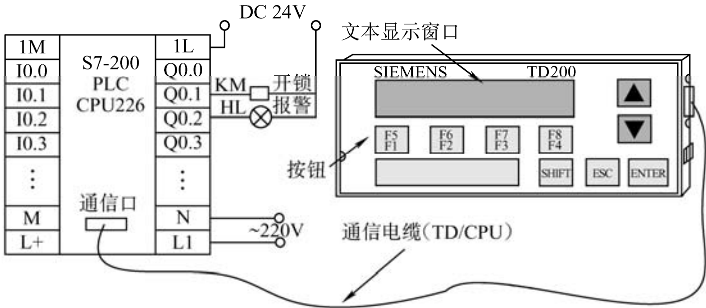
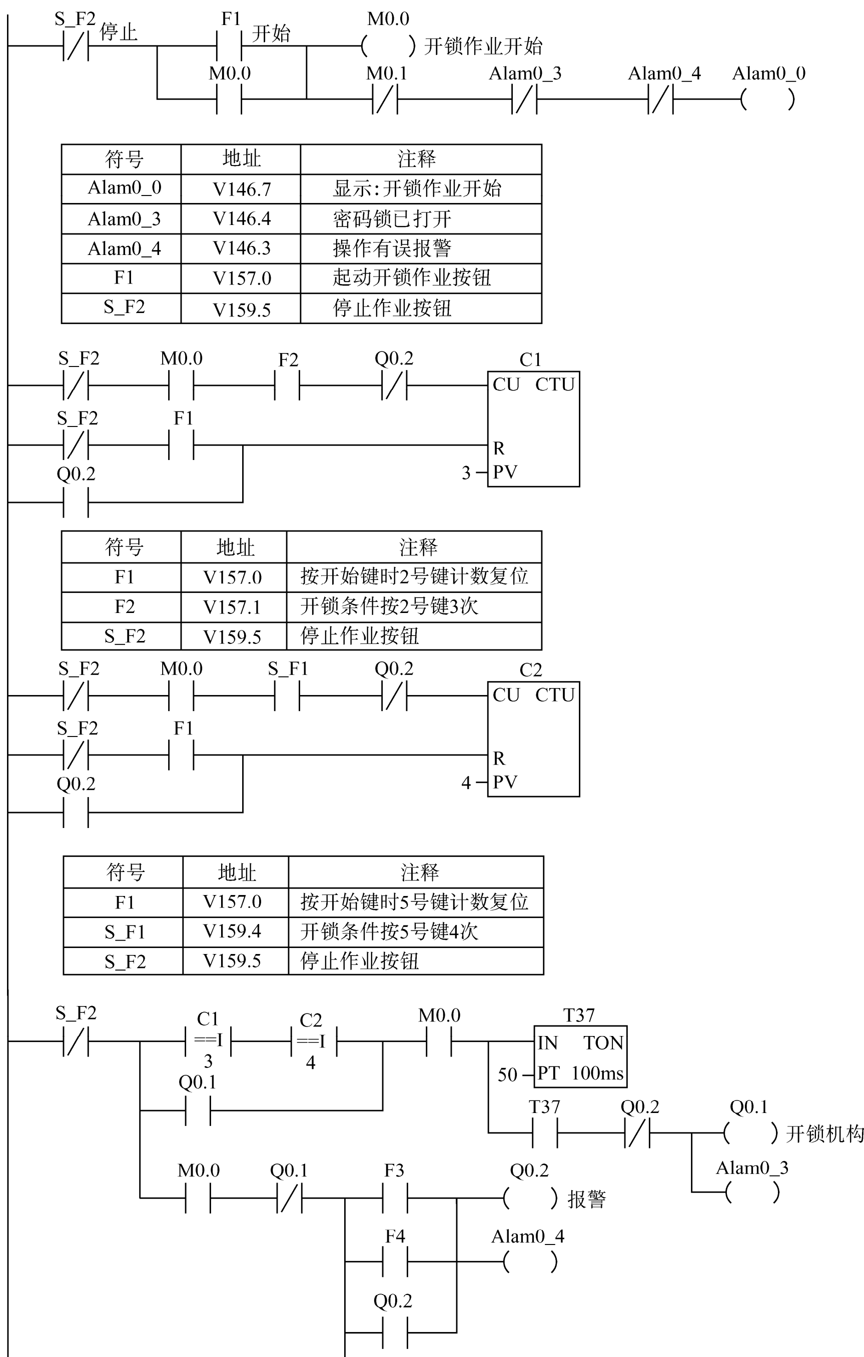


图 4-64　PLC 与 TD 连接示意及 PLC 对外接线图

（2）程序设计及注释如下：

密码锁开启梯形图如图 4-65 所示。

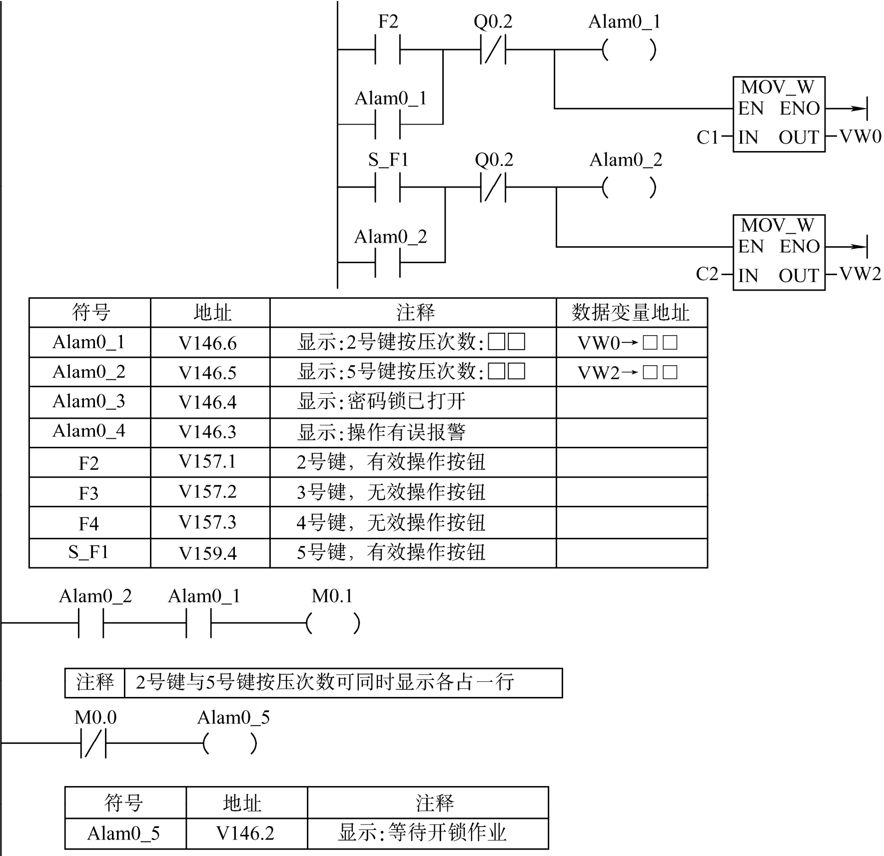


符号	地址	注释
Alam0_0	V146.7	显示:开锁作业开始
Alam0_3	V146.4	密码锁已打开
Alam0_4	V146.3	操作有误报警
F1	V157.0	起动开锁作业按钮
S_F2	V159.5	停止作业按钮

符号	地址	注释
F1	V157.0	按开始键时2号键计数复位
F2	V157.1	开锁条件按2号键3次
S_F2	V159.5	停止作业按钮

符号	地址	注释
F1	V157.0	按开始键时5号键计数复位
S_F1	V159.4	开锁条件按5号键4次
S_F2	V159.5	停止作业按钮

图 4-65　密码锁开启梯形图及注释



符号	地址	注释	数据变量地址
Alam0_1	V146.6	显示:2号键按压次数:□□	VW0→□□
Alam0_2	V146.5	显示:5号键按压次数:□□	VW2→□□
Alam0_3	V146.4	显示:密码锁已打开	
Alam0_4	V146.3	显示:操作有误报警	
F2	V157.1	2号键，有效操作按钮	
F3	V157.2	3号键，无效操作按钮	
F4	V157.3	4号键，无效操作按钮	
S_F1	V159.4	5号键，有效操作按钮	

注释	2号键与5号键按压次数可同时显示各占一行

符号	地址	注释
Alam0_5	V146.2	显示:等待开锁作业

图 4-65 密码锁开启梯形图及注释（续）

例十九、变频器控制电动机实现 15 段速运转控制

PLC 与变频器在当今控制系统中都是主控器件，一个作为控制部分的核心；一个作为拖动部分的核心，很多生产设备都是二者配合组成控制系统，应用 PLC 重在编程；应用变频器重在参数设定。本例使用西门子 MM440 型变频器，实现电动机的多段速控制。

1. 控制要求

按下起动按钮，电动机从零速开始起动运行到 5Hz；在此频率运行 10s 后，频率改为 8Hz；在 8Hz 运行 10s 后，频率改为 11Hz；就这样每隔 10s 电动机运行频率提高 3Hz，形成 5→8→11→14→17→20→23→26→29→32→35→38→41→44→47（Hz），共计 15 个速度段。什么时候电动机需停止，按下停止按钮即可。

2. MM440 变频器简介

变频器的输入电源可接交流三相或单相，输出接三相交流电动机，电动机功率从一百多瓦到二百千瓦以上，依据电动机选变频器。除了主线路还有控制线路，控制线路也分输入信号与输出信号，为了让变频器工作就要给它输入信号，输入信号分并行口和串行口，并行口又分数字量和模拟量。因本例只涉及数字量输入信号，所以对变频器简介到此。

3. MM440 变频器的数字量输入端口

有 8 个端口可作为数字量输入，本例使用它的“5”、“6”、“7”、“8”、“16”共 5 个端口，数字量即开关量，PLC 的输出信号刚好作为变频器的输入信号，触点闭合即为有信号，有信号为逻辑“1”，无信号为逻辑“0”。本例变频器要有 15 个频率输出，用“5”、“6”、“7”、“8”四个端口的状态组合，刚好能形成 15 个信号，“16”号端口用作启/停信号。某个端口是否可参与状态组合，这就要进行参数设置了，负责端口 5 的参数是 P0701；端口 6 是 P0702；7 是 P0703；8 是 P0704；16 是 P0705。每个参数都有几个可选项，选择一个符合要求的就可以了。该口的功能设置好后还要找到与端口或端口组合相对应的频率放在哪里了，这组参数就是 P1001 到 P1015，端口与参数及其频率值的对应关系如表 4-3 所示。

表 4-3　数字量输入端口与参数及其频率值对应关系

参数号	频率（Hz）	端口 5	端口 6	端口 7	端口 8	端口 16
P1001	5	1	0	0	0	1
P1002	8	0	1	0	0	1
P1003	11	1	1	0	0	1
P1004	14	0	0	1	0	1
P1005	17	1	0	1	0	1
P1006	20	0	1	1	0	1
P1007	23	1	1	1	0	1
P1008	26	0	0	0	1	1
P1009	29	1	0	0	1	1
P1010	32	0	1	0	1	1
P1011	35	1	1	0	1	1
P1012	38	0	0	1	1	1
P1013	41	1	0	1	1	1
P1014	44	0	1	1	1	1
P1015	47	1	1	1	1	1

4. PLC 与变频器之间的线路连接

首先要进行 PLC 的 I/O 分配：起动按钮 SB1 的常开触点接 I0.0；停止按钮 SB2 的常闭触点接 I0.1；Q0.0 接变频器的端口 5；Q0.1 接端口 6；Q0.2 接端口 7；Q0.3 接端口 8；Q0.4 接端口 16。控制电路图如图 4-66 所示。

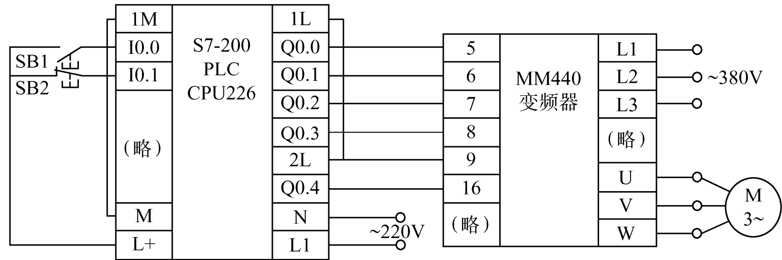


图 4-66 PLC 与变频器联机控制电路图

5. 变频器参数设置

使用变频器首先要把对外端口的线路接好，特别是输出接电动机的线与输入电源线一定不能接混。然后就是参数设置了，表 4-4 列出与本例有关的参数设置，在此不包括与电动机有关的参数设置以及参数的出厂值恢复过程。

表 4-4 实现 15 段速控制变频器参数设置

参数号	设置值	说明	参数号	设置值	说明
P0003	3	用户访问为专家级	P1003	11	设置固定频率 3
P0004	7	命令和数字量 I/O	P1004	14	设置固定频率 4
P0010	1（0）	设参数时为 1 运行为 0	P1005	17	设置固定频率 5
P0700	2	由端口信号控制运行	P1006	20	设置固定频率 6
P0701	17	端口 5 为二进制组合	P1007	23	设置固定频率 7
P0702	17	端口 6 为二进制组合	P1008	26	设置固定频率 8
P0703	17	端口 7 为二进制组合	P1009	29	设置固定频率 9
P0704	17	端口 8 为二进制组合	P1010	32	设置固定频率 10
P0705	1	端口 16 为控制起/停	P1011	35	设置固定频率 11
P0100	0	供电线路频率为 50Hz	P1012	38	设置固定频率 12
P1000	3	数字量输入的固定频率	P1013	41	设置固定频率 13
P1001	5	设置固定频率 1	P1014	44	设置固定频率 14
P1002	8	设置固定频率 2	P1015	47	设置固定频率 15

6. 程序设计

两种设计方案，方案 1 的梯形图程序如图 4-67 所示，方案 2 如图 4-68 所示。

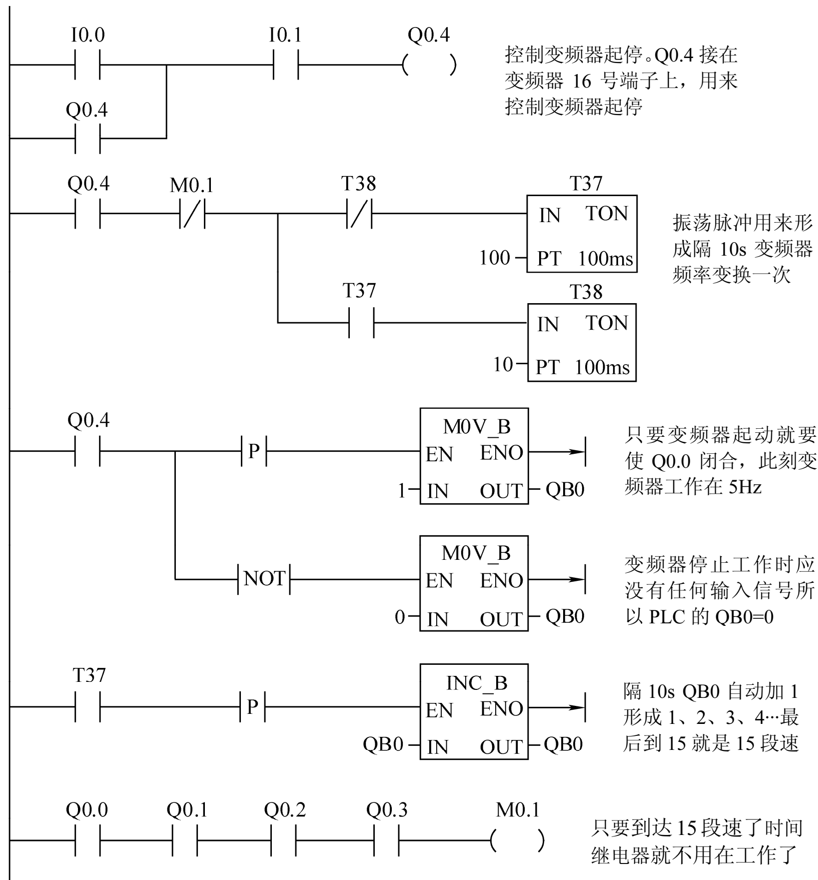


图 4-67　电动机实现 15 段速运转 PLC 程序梯形图 1

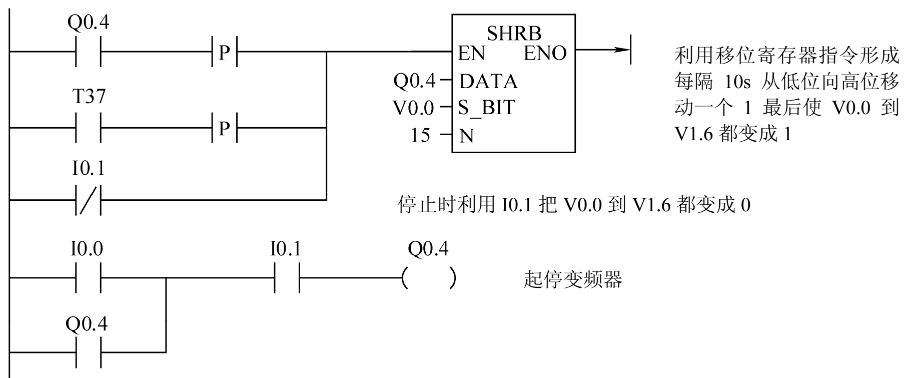


图 4-68　电动机实现 15 段速运转 PLC 程序梯形图 2

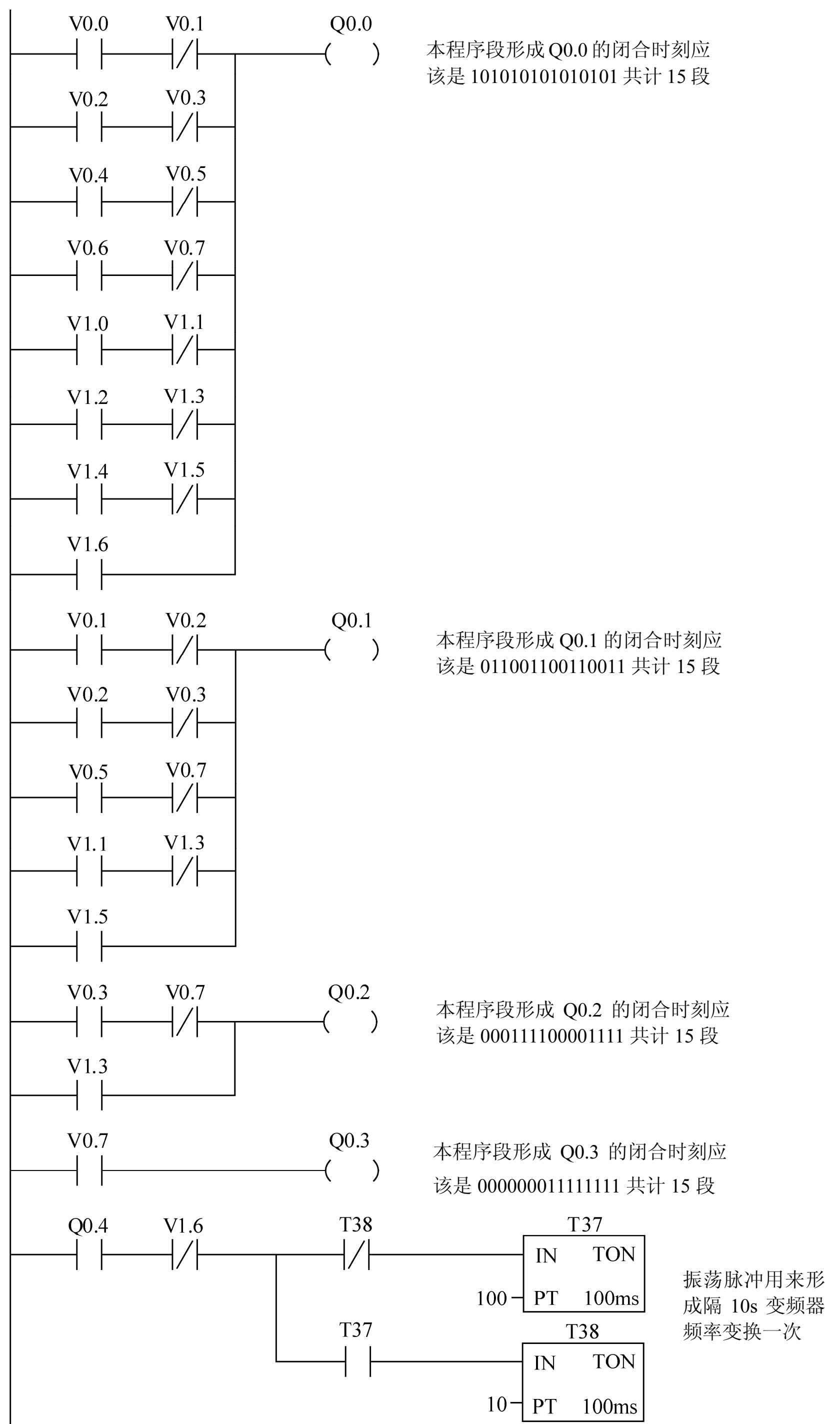


图 4-68 电动机实现 15 段速运转 PLC 程序梯形图 2（续）

例二十、饮料自动售货机控制程序

1. 控制要求

① 自动售货机可投入 1 角、5 角、1 元的硬币（硬币识别装置另议，这里只作为输入信号）；

② 当投入的硬币总值超过 2.5 元时，啤酒指示灯亮；当投入的硬币总值超过 4 元时，橙汁及啤酒指示灯都亮；

③ 当啤酒指示灯亮时，按动放啤酒按钮，则放出啤酒，6s 后自动停止。在此过程中啤酒指示灯闪烁；

④ 当橙汁指示灯亮时，按动放橙汁按钮，则放出橙汁，6s 后自动停止。在此过程中橙汁指示灯闪烁；

⑤ 当已开始放出某种饮料时，系统自动计算所剩钱币，如小于 2.5 元时啤酒与橙汁指示灯全灭。如大于等于 2.5 元而小于 4 元时，啤酒指示灯继续亮，而大于等于 4 元时啤酒及橙汁指示灯都亮；

⑥ 投完硬币不喝饮料不能将钱币退出，然后经 20s 延时没再选择放出饮料或按下找钱按钮可将剩余硬币退出。

2. 程序设计

本题使用了一些数学计算指令，钱币的识别机构本题不考虑。

（1）饮料自动售货机结构示意图如图 4-69 所示。

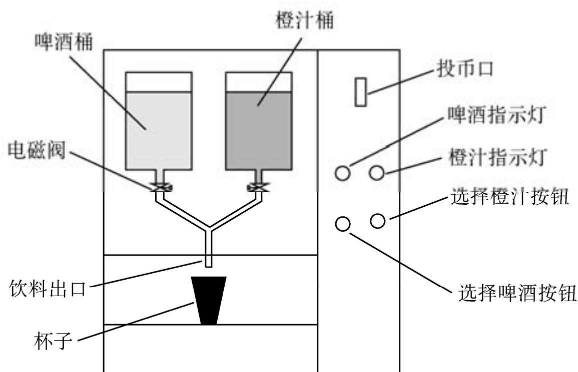


图 4-69　饮料自动售货机结构示意图

（2）饮料自动售货机 PLC 控制接线图如图 4-70 所示。

（3）控制系统的语句表程序及注释如下：

```
主程序                      // 程序注释
Network 1                   // 将硬币基数放入对应存储器中初始化段
LD        SM0.1
```

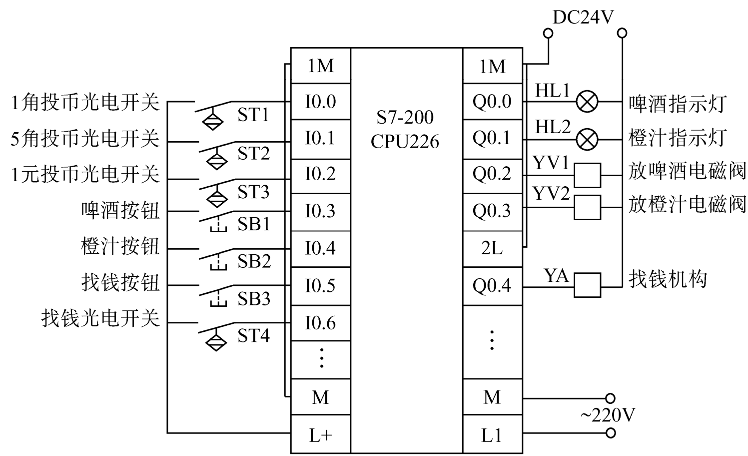


图 4-70 饮料自动售货机 PLC 控制接线图

```
MOVW   1, MW0            // 1 角硬币基数放入存储器中
MOVW   5, MW2            // 5 角硬币基数放入存储器中
MOVW   10, MW4           // 1 元硬币基数放入存储器中
Network 2                // 钱币识别段
LD     SM0.0
LPS
A      I0.0              // 1 角硬币识别口
EU
+I     MW0, MW10         // 把送进来的 1 角钱放入总存储器中
LRD
A      I0.1              // 5 角硬币识别口
EU
+I     MW2, MW10         // 把送进来的 5 角钱放入总存储器中
LPP
A      I0.2              // 1 元硬币识别口
EU
+I     MW4, MW10         // 把送进来的 1 元钱放入总存储器中
Network 3                // 钱币比较段
LDW>=  MW10, 25          // 把送进来的钱与 2 元 5 角进行比较是否具备放
LD     Q0.2              // 啤酒的条件
OW<    MW10, 25
NOT
```

```
A          M6.0
OLD
=          M6.0             // 具备放啤酒条件时此继电器闭合
Network 4                   // 钱币比较段
LDW>=      MW10, 40         // 把送进来的钱与 4 元 0 角进行比较是否具备放
LD         Q0.3             // 橙汁的条件
OW<        MW10, 40
NOT
A          M6.1
OLD
=          M6.1             // 具备放橙汁条件时此继电器闭合
Network 5                   // 放饮料操作程序段
LD         SM0.0
LPS
LD         M6.0
AN         Q0.2
LD         Q0.2             // 一旦开始放啤酒相应的指示灯闪烁
A          SM0.5
OLD
ALD
=          Q0.0             // 放啤酒条件具备时相应指示灯亮
LRD
AN         Q0.4             // 在没有找钱操作的前提下可以放啤酒
A          Q0.0             // 在指示灯已亮的前提下可以放啤酒
A          I0.3             // 放啤酒按钮
EU
S          Q0.2, 1          // 打开放啤酒的电磁阀
-I         25, MW10         // 将总钱币减去 2 元 5 角
LRD
LPS
A          Q0.2
TON        T37, 60          // 打开电磁阀放啤酒的时间为 6s
LPP
A          T37
R          Q0.2, 1          // 6s 后关闭放啤酒的电磁阀
```

```
LRD
LD      M6.1
AN      Q0.3
LD      Q0.3                    // 一旦开始放橙汁，相应的指示灯闪烁
A       SM0.5
OLD
ALD
=       Q0.1                    // 放橙汁条件具备时相应指示灯亮
LRD
AW>=    MW10, 40                // 总钱币大于 4 元才能操作放橙汁按钮
AN      Q0.4                    // 没有找钱操作才能操作放橙汁按钮
A       Q0.1
A       I0.4                    // 放橙汁按钮
EU
S       Q0.3, 1                 // 打开放橙汁的电磁阀
–I      40, MW10                // 将总钱币减去 4 元 0 角
LPP
LPS
A       Q0.3
TON     T38, 60                 // 打开电磁阀放橙汁的时间为 6s
LPP
A       T38
R       Q0.3, 1                 //  6s 后关闭放橙汁的电磁阀
Network 6                       // 如没有操作放饮料经延时后可以找钱程序段
LDW>=   MW10, 25
AN      M1.3
TON     T39, 200                // 没有操作放饮料 20s 后可找钱
Network 7
LD      I0.3                    // 一旦有放饮料操作，找钱动作将不能继续
O       I0.4
LDN     Q0.2
AN      Q0.3
NOT
LPS
A       M1.3
```

```
=       M1.3            // 在没到 20s 前操作放饮料按钮仍然有效
LPP
ALD
O       M1.3
=       M1.3
Network 8               // 自动或手动找钱程序段
LD      T39             // 延时后自动找钱
O       I0.5            // 手动找钱
AN      Q0.2            // 不能有放啤酒动作
AN      Q0.3            // 不能有放橙汁动作
S       Q0.4, 1         // 找钱机构动作
Network 9               // 找出钱币与总内存钱数比较程序段
LD      Q0.4
A       I0.6            // 找钱数量记录
LDN     Q0.4
CTU     C10, MW10       // 找出与内存比较
Network 10              // 清空存钱盒程序段
LD      C10             // 所有的钱币都找出去了
MOVW    0, MW10         // 清空存钱盒
Network 11              // 关闭找钱机构
LD      C10             // 所有的钱币都找出去了
O       I0.3            // 或有放啤酒按钮操作
O       I0.4            // 或有放橙汁按钮操作
R       Q0.4, 1         // 找钱机构关闭
```

第五章 系统设计实例

系统设计的题目比一般实例的难度要大些，是在已有 PLC 的理论知识之后，有一些编写实例的基础，又对某个实际应用环节有所了解，所进行的较为完整的综合性设计。

例一、百天倒计时控制程序

本题以北京 2008 年奥运会倒计时为例，说明倒计时控制器也可由 PLC 来控制。该倒计时器可动态地显示天、小时、分、秒，开机后显示初始状态（全零态），然后按当天距开幕的实际天数进行设置。设置好后按下起动按钮，百天倒计时就开始了，如按停止按钮将停止计时，显示当前状态。天、小时、分、秒各状态可随时调整与设置，各状态显示采用动态循环扫描方式。

1. 控制要求

（1）开机时初始状态显示为 00 天 00 时 00 分 00 秒；

（2）按下起动按钮（SB1），倒计时开始，1s 后显示为 99 天 23 时 59 分 59 秒；

（3）可根据当时的时间调整至距开幕的实际时间，按住停止按钮（SB2）2s 后，即转为调整期，显示天数的数码闪烁，这时可进行天数的调整，调整好天数按一次停止按钮即转为小时的调整，显示小时的数码闪烁，调整好小时按一下停止按钮即转为分钟的调整，以此类推直到调整完秒，调整过程结束。按下起动按钮（SB1）即可计时。

（4）时间的调整可进行上调或下调，在调整期按上调按钮（SB3），按一次升 1 个数，按下调按钮（SB4），按一次降 1 个数。

（5）可任意停止于任意时间状态，在停止状态可任意调整时间。

（6）各时间状态采用动态循环扫描，也就是说每一时刻只有一组七段数码管在通电但肉眼是看不出来的，这样做是为了节省 PLC 的输出点数。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，本题使用了许多数学运算指令，难点在于倒计时的减进位及动态扫描，所谓动态扫描就是利用 PLC 周期性扫描的特点，在编程时要做到每个周期只有一组数码能够形成通电回路，从而得电点亮，因一个扫描周期的时间相当短，只有几十毫秒，所以感觉每个灯都在通电亮着。本题共有 8 只数码管，按“天、时、分、秒”来分，需分成 4 组，每组 2 只，也就是说某一时刻只有 2 只数码管得电。倒计时控制器示意图如图 5-1

所示。

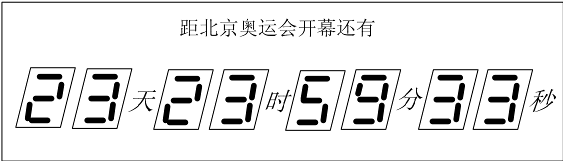


图 5-1　倒计时控制器示意图

（2）本题所使用的 PLC 是 CPU226 晶体管输出型，8 只数码管的公共端分别接在 Q0.0～Q0.7 端子上。倒计时控制器 PLC 接线图如图 5-2 所示。

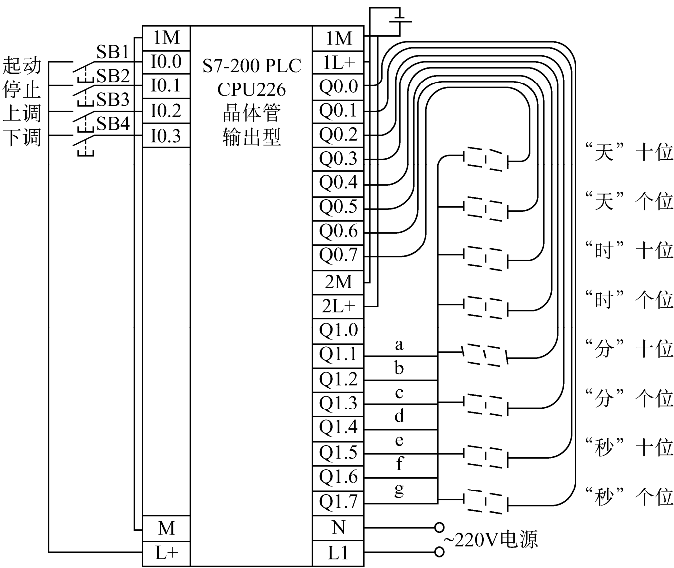


图 5-2　倒计时控制器的 PLC 控制接线图

（3）倒计时控制器的 PLC 控制程序的语句表及注释如下：

```
主程序                                    程序注释
Network 1                          //  正常起动与调整设置程序段
LD        SM0.0
LPS
A         I0.1                     //  按住停止按钮 2s 将转为调整设置
AN        M0.0                     //  在没有正常计时的情况下才能调整
TON       T37, 20                  //  2s 定时器
LRD
LD        T37
```

```
O         M0.1
ALD
LPS
AN        I0.0          // 在调整过程中只要按起动按钮调整即刻结束
=         M0.1          // 定时时间到后即可转为调整设置工作
LRD
AW<       C0, 1         // 调整秒数的必要条件
=         M0.4          // 条件满足此继电器闭合
LRD
AW=       C0, 1         // 调整分钟数的必要条件
=         M0.5          // 条件满足此继电器闭合
LRD
AW=       C0, 2         // 调整小时数的必要条件
=         M0.6          // 条件满足此继电器闭合
LPP
AW=       C0, 3         // 调整天数的必要条件
=         M0.7          // 条件满足此继电器闭合
LPP
LD        I0.0          // 按起动按钮，倒计时开始工作
O         M0.0
ALD
LPS
AN        I0.1          // 按下停止按钮正常计时即刻结束
AN        M0.2          // 正常倒计时到终了时刻此点断开
=         M0.0          // 此继电器闭合既是正常倒计时
LPP
R         M0.2, 2       // 再一次起动时断开终了继电器
Network 2               // 调整设置程序段
LD        M0.1          // 在调整设置工作段期间
A         I0.1          // 按一次调整按钮调整段变化一次
EU
LD        M0.0          // 当转为正常倒计时工作段时将计数器复位
CTU       C0, 4         // 在调整设置工作段决定调整段的计数器
Network 3               // 决定倒计时的基本时间程序段
LD        M0.0          // 决定正常倒计时工作段的继电器必须闭合
```

```
AN      M0.1        // 决定调整设置工作段的继电器必须断开
LPS
AN      T39
TON     T38, 5
LPP
A       T38
TON     T39, 5      // 基本定时时间定为 1s
Network 4           // 决定秒数位变化的程序段
LD      SM0.0
LPS
AW<     MW1, 1      // 当秒数个位中的数据小于 1
MOVW    1, MW1      // 将数据 1 送入秒数个位中使之显示为 0
LRD
LDW<    C0, 1       // 在此条件下调整秒数
A       M0.1        // 必须在调整中设置工作段
A       I0.3        // 在调整中设置工作段，减少秒数
O       T39         // 正常倒计时的秒数变化基准时间
ALD
EU
SLW     MW1, 1      // 每 1s 或每调整一次使秒数个位中的数据减 1
LPP
LPS
AW<     C0, 1       // 在此条件下调整秒数
A       M0.1        // 必须在调整中设置工作段
A       I0.2        // 在调整中设置工作段增加秒数
EU
SRW     MW1, 1      // 每调整一次使秒数个位中的数据加 1
LRD
A       M2.2        // 当秒数个位倒计时到此时
R       M1.1, 10    // 将秒数个位复位为 0
LRD
AB<     MB3, 1      // 当秒数十位的数据小于 1h
MOVB    1, MB3      // 将数据 1 送入秒数十位中使之显示为 0
LRD
A       M1.1        // 只要秒数个位为 9，十位就自动减 1
```

```
EU
SLB     MB3, 1          // 秒数十位减 1
LPP
A       M3.6            // 当秒数十位倒计时到此时
R       M3.1, 6         // 将秒数十位复位为 0
Network 5               // 决定分钟数位变化的程序段
LD      SM0.0
LPS
AW<     MW4, 1          // 当分钟数个位中的数据小于 1
MOVW    1, MW4          // 将数据 1 送入分钟数个位中去
LRD
LDW=    C0, 1           // 在此条件下调整分钟数
A       M0.1            // 必须在调整中设置工作段
A       I0.3            // 在调整设置工作段减少分钟数
O       M3.1            // 正常倒计时的分钟数个位变化继电器
ALD
EU
SLW     MW4, 1          // 每 1min 或每调整一次使分钟数个位中的数据减 1
LPP
LPS
AW=     C0, 1           // 在此条件下调整分钟数
A       M0.1            // 必须在调整中设置工作段
A       I0.2            // 在调整设置工作段增加分钟数
EU
SRW     MW4, 1          // 每调整一次使分钟数个位中的数据加 1
LRD
A       M5.2            // 当分钟数个位倒计时到此时
R       M4.1, 10        // 使分钟个位数复位为 0
LRD
AB<     MB6, 1          // 当分钟数十位的数据小于 1h
MOVB    1, MB6          // 将数据 1 送入分钟数十位中使之显示为 0
LRD
A       M4.1            // 只要分钟数个位为 9，十位就自动减 1
EU
SLB     MB6, 1          // 分钟数十位减 1
```

```
LPP
A       M6.6          // 当分钟数十位倒计时到此时
R       M6.1, 6       // 将分钟数十位复位为 0
Network 6             // 决定小时数位变化的程序段
LD      SM0.0
LPS
AD<     MD7, 1        // 当小时位中的数据小于 1
MOVD    1, MD7        // 将数据 1 送入小时位中去
LRD
LDW=    C0, 2         // 在此条件下调整小时数
A       M0.1          // 必须在调整设置工作段
A       I0.3          // 在调整设置工作段减少小时数
O       M6.1          // 正常倒计时的小时数变化继电器
ALD
EU
SLD     MD7, 1        // 每 1h 或每调整一次使小时数减 1
LPP
LPS
AW=     C0, 2         // 在此条件下调整小时数
A       M0.1          // 必须在调整中设置工作段
A       I0.2          // 在调整设置工作段增加小时数
EU
SRD     MD7, 1        // 每调整一次使小时数中的数据加 1
LPP
A       M10.0         // 当小时数倒计时到此时
R       M7.1, 24      // 将小时数复位为 0
Network 7             // 决定天数数位变化的程序段
LD      SM0.0
LPS
AW<     MW11, 1       // 当天数个位中的数据小于 1
MOVW    1, MW11       // 将数据 1 送入天数个位中去
LRD
LDW=    C0, 3         // 在此条件下调整天数
A       M0.1          // 必须在调整中设置工作段
A       I0.3          // 在调整设置工作段减少天数
```

```
O        M7.1          // 正常倒计时的天数个位变化继电器
ALD
EU
SLW      MW11, 1       // 每一天或每调整一次使天数个位中的数据减 1
LPP
LPS
AW=      C0, 3         // 在此条件下调整天数
A        M0.1          // 必须在调整中设置工作段
A        I0.2          // 在调整设置工作段增加天数
EU
SRW      MW11, 1       // 每调整一次使天数个位中的数据加 1
LRD
A        M12.2         // 当天数个位倒计时到此时
R        M11.1, 10     // 使天数个位数复位为 0
LRD
AW<      MW13, 1       // 当天数十位的数据小于 1 时
MOVW     1, MW13       // 将数据 1 送入天数十位中使之显示为 0
LRD
A        M11.1         // 只要天数个位为 9，十位就自动减 1
EU
SLW      MW13, 1       // 天数十位减 1
LPP
A        M14.2         // 当天数十位倒计时到此时
R        M13.1, 10     // 将天数十位复位为 0
Network 8              // 正常倒计时接近终了时刻的锁定程序段
LD       M3.0          // 秒的十位已计到 0
A        M4.0          // 分钟的个位已计到 0
A        M6.0          // 分钟的十位已计到 0
A        M7.0          // 小时已计到 0
A        M11.0         // 天数的个位已计到 0
A        M13.0         // 天数的十位已计到 0
A        M1.5          // 秒的个位已倒计到 5s
S        M0.3, 1       // 此时让 M0.3 继电器得电锁定
Network 9              // 正常倒计时终了时刻的锁定程序段
LD       M0.3          // 倒计时到 5s 时的锁定信号
```

```
A        M1.0          // 到此刻已倒计时到 0
S        M0.2, 1       // 立即形成一个锁定信号停止全部计时
Network 10             // 七段数码管显示“9”的程序段
LD       M0.4          // 在调整设置秒数时接通
A        SM0.5         // 在调整设置秒数时使数码管闪烁
ON       M0.4          // 在正常倒计时工作段接通
A        M1.1          // 接通时秒数个位显示“9”
A        Q0.0          // 秒数个位七段数码管的公共端
LD       M0.5          // 在调整设置分钟时接通
A        SM0.5         // 在调整设置分钟时使数码管闪烁
ON       M0.5          // 在正常倒计时工作段接通
A        M4.1          // 接通时分钟数个位显示“9”
A        Q0.2          // 分钟数个位七段数码管的公共端
OLD
LD       M0.6          // 在调整设置小时时接通
A        SM0.5         // 在调整设置小时时使数码管闪烁
ON       M0.6          // 在正常倒计时工作段接通
LD       M7.5          // 接通时小时显示“19”
O        M8.7          // 接通时小时显示“9”
ALD
A        Q0.4          // 小时个位七段数码管的公共端
OLD
LD       M0.7          // 在调整设置天数时接通
A        SM0.5         // 在调整设置天数时使数码管闪烁
ON       M0.7          // 在正常倒计时工作段接通
LD       M11.1         // 接通时天数个位显示“9”
A        Q0.6          // 天数个位七段数码管的公共端
LD       M13.1         // 接通时天数十位显示“9”
A        Q0.7          // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB     16#DE, QB1    // 将数字“9”送到输出端通过七段数码管显示
Network 11             // 七段数码管显示“8”的程序段
LD       M0.4          // 在调整设置秒数时接通
```

```
A       SM0.5          // 在调整设置秒数时使数码管闪烁
ON      M0.4           // 在正常倒计时工作段接通
A       M1.2           // 接通时秒数个位显示“8”
A       Q0.0           // 秒数个位七段数码管的公共端
LD      M0.5           // 在调整设置分钟时接通
A       SM0.5          // 在调整设置分钟时使数码管闪烁
ON      M0.5           // 在正常倒计时工作段接通
A       M4.2           // 接通时分钟个位显示“8”
A       Q0.2           // 分钟个位七段数码管的公共端
OLD
LD      M0.6           // 在调整设置小时时接通
A       SM0.5          // 在调整设置小时时使数码管闪烁
ON      M0.6           // 在正常倒计时工作段接通
LD      M7.6           // 接通时小时显示“18”
O       M9.0           // 接通时小时显示“8”
ALD
A       Q0.4           // 小时个位七段数码管的公共端
OLD
LD      M0.7           // 在调整设置天数时接通
A       SM0.5          // 在调整设置天数时使数码管闪烁
ON      M0.7           // 在正常倒计时工作段接通
LD      M11.2          // 接通时天数个位显示“8”
A       Q0.6           // 天数个位七段数码管的公共端
LD      M13.2          // 接通时天数十位显示“8”
A       Q0.7           // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB    16#FE, QB1     // 将数字“8”送到输出端通过七段数码管显示
Network 12             // 七段数码管显示“7”的程序段
LD      M0.4           // 在调整设置秒数时接通
A       SM0.5          // 在调整设置秒数时使数码管闪烁
ON      M0.4           // 在正常倒计时工作段接通
A       M1.3           // 接通时秒数个位显示“7”
A       Q0.0           // 秒数个位七段数码管的公共端
```

```
LD      M0.5            // 在调整设置分钟时接通
A       SM0.5           // 在调整设置分钟时使数码管闪烁
ON      M0.5            // 在正常倒计时工作段接通
A       M4.3            // 接通时分钟个位显示“7”
A       Q0.2            // 分钟个位七段数码管的公共端
OLD
LD      M0.6            // 在调整设置小时时接通
A       SM0.5           // 在调整设置小时时使数码管闪烁
ON      M0.6            // 在正常倒计时工作段接通
LD      M7.7            // 接通时小时显示“17”
O       M9.1            // 接通时小时显示“7”
ALD
A       Q0.4            // 小时个位七段数码管的公共端
OLD
LD      M0.7            // 在调整设置天数时接通
A       SM0.5           // 在调整设置天数时使数码管闪烁
ON      M0.7            // 在正常倒计时工作段接通
LD      M11.3           // 接通时天数个位显示“7”
A       Q0.6            // 天数个位七段数码管的公共端
LD      M13.3           // 接通时天数十位显示“7”
A       Q0.7            // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB    16#0E, QB1      // 将数字“7”送到输出端通过七段数码管显示
Network 13              // 七段数码管显示“6”的程序段
LD      M0.4            // 在调整设置秒数时接通
A       SM0.5           // 在调整设置秒数时使数码管闪烁
ON      M0.4            // 在正常倒计时工作段接通
A       M1.4            // 接通时秒数个位显示“6”
A       Q0.0            // 秒数个位七段数码管的公共端
LD      M0.5            // 在调整设置分钟时接通
A       SM0.5           // 在调整设置分钟时使数码管闪烁
ON      M0.5            // 在正常倒计时工作段接通
A       M4.4            // 接通时分钟个位显示“6”
```

```
A        Q0.2           // 分钟个位七段数码管的公共端
OLD
LD       M0.6           // 在调整设置小时时接通
A        SM0.5          // 在调整设置小时时使数码管闪烁
ON       M0.6           // 在正常倒计时工作段接通
LD       M8.0           // 接通时小时显示“16”
O        M9.2           // 接通时小时显示“6”
ALD
A        Q0.4           // 小时个位七段数码管的公共端
OLD
LD       M0.7           // 在调整设置天数时接通
A        SM0.5          // 在调整设置天数时使数码管闪烁
ON       M0.7           // 在正常倒计时工作段接通
LD       M11.4          // 接通时天数个位显示“6”
A        Q0.6           // 天数个位七段数码管的公共端
LD       M13.4          // 接通时天数十位显示“6”
A        Q0.7           // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB     16#FA, QB1     // 将数字“6”送到输出端通过七段数码管显示
Network 14              // 七段数码管显示“5”的程序段
LD       M0.4           // 在调整设置秒数时接通
A        SM0.5          // 在调整设置秒数时使数码管闪烁
ON       M0.4           // 在正常倒计时工作段接通
LD       M1.5           // 接通时秒数个位显示“5”
A        Q0.0           // 秒数个位七段数码管的公共端
LD       M3.1           // 接通时秒数十位显示“5”
A        Q0.1           // 秒数十位七段数码管的公共端
OLD
ALD
LD       M0.5           // 在调整设置分钟时接通
A        SM0.5          // 在调整设置分钟时使数码管闪烁
ON       M0.5           // 在正常倒计时工作段接通
LD       M4.5           // 接通时分钟个位显示“5”
```

```
A        Q0.2           // 分钟个位七段数码管的公共端
LD       M6.1           // 接通时分钟十位显示“5”
A        Q0.3           // 分钟十位七段数码管的公共端
OLD
ALD
OLD
LD       M0.6           // 在调整设置小时时接通
A        SM0.5          // 在调整设置小时时使数码管闪烁
ON       M0.6           // 在正常倒计时工作段接通
LD       M8.1           // 接通时小时显示“15”
O        M9.3           // 接通时小时显示“5”
ALD
A        Q0.4           // 小时个位七段数码管的公共端
OLD
LD       M0.7           // 在调整设置天数时接通
A        SM0.5          // 在调整设置天数时使数码管闪烁
ON       M0.7           // 在正常倒计时工作段接通
LD       M11.5          // 接通时天数个位显示“5”
A        Q0.6           // 天数个位七段数码管的公共端
LD       M13.5          // 接通时天数十位显示“5”
A        Q0.7           // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB     16#DA, QB1     // 将数字“5”送到输出端通过七段数码管显示
Network 15              // 七段数码管显示“4”的程序段
LD       M0.4           // 在调整设置秒数时接通
A        SM0.5          // 在调整设置秒数时使数码管闪烁
ON       M0.4           // 在正常倒计时工作段接通
LD       M1.6           // 接通时秒数个位显示“4”
A        Q0.0           // 秒数个位七段数码管的公共端
LD       M3.2           // 接通时秒数十位显示“4”
A        Q0.1           // 秒数十位七段数码管的公共端
OLD
ALD
```

```
LD      M0.5            // 在调整设置分钟时接通
A       SM0.5           // 在调整设置分钟时使数码管闪烁
ON      M0.5            // 在正常倒计时工作段接通
LD      M4.6            // 接通时分钟个位显示“4”
A       Q0.2            // 分钟个位七段数码管的公共端
LD      M6.2            // 接通时分钟十位显示“4”
A       Q0.3            // 分钟十位七段数码管的公共端
OLD
ALD
OLD
LD      M0.6            // 在调整设置小时时接通
A       SM0.5           // 在调整设置小时时使数码管闪烁
ON      M0.6            // 在正常倒计时工作段接通
LD      M8.2            // 接通时小时显示“14”
O       M9.4            // 接通时小时显示“4”
ALD
A       Q0.4            // 小时个位七段数码管的公共端
OLD
LD      M0.7            // 在调整设置天数时接通
A       SM0.5           // 在调整设置天数时使数码管闪烁
ON      M0.7            // 在正常倒计时工作段接通
LD      M11.6           // 接通时天数个位显示“4”
A       Q0.6            // 天数个位七段数码管的公共端
LD      M13.6           // 接通时天数十位显示“4”
A       Q0.7            // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB    16#CC, QB1      // 将数字“4”送到输出端通过七段数码管显示
Network 16              // 七段数码管显示“3”的程序段
LD      M0.4            // 在调整设置秒数时接通
A       SM0.5           // 在调整设置秒数时使数码管闪烁
ON      M0.4            // 在正常倒计时工作段接通
LD      M1.7            // 接通时秒数个位显示“3”
A       Q0.0            // 秒数个位七段数码管的公共端
```

```
LD      M3.3            // 接通时秒数十位显示“3”
A       Q0.1            // 秒数十位七段数码管的公共端
OLD
ALD
LD      M0.5            // 在调整设置分钟时接通
A       SM0.5           // 在调整设置分钟时使数码管闪烁
ON      M0.5            // 在正常倒计时工作段接通
LD      M4.7            // 接通时分钟个位显示“3”
A       Q0.2            // 分钟个位七段数码管的公共端
LD      M6.3            // 接通时分钟十位显示“3”
A       Q0.3            // 分钟十位七段数码管的公共端
OLD
ALD
OLD
LD      M7.1            // 接通时小时显示“23”
O       M8.3            // 接通时小时显示“13”
O       M9.5            // 接通时小时显示“3”
LD      M0.6            // 在调整设置小时时接通
A       SM0.5           // 在调整设置小时时使数码管闪烁
ON      M0.6            // 在正常倒计时工作段接通
ALD
A       Q0.4            // 小时个位七段数码管的公共端
OLD
LD      M0.7            // 在调整设置天数时接通
A       SM0.5           // 在调整设置天数时使数码管闪烁
ON      M0.7            // 在正常倒计时工作段接通
LD      M11.7           // 接通时天数个位显示“3”
A       Q0.6            // 天数个位七段数码管的公共端
LD      M13.7           // 接通时天数十位显示“3”
A       Q0.7            // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB    16#9E, QB1      // 将数字“3”送到输出端通过七段数码管显示
Network 17              // 七段数码管显示“2”的程序段
```

```
LD     M0.4          // 在调整设置秒数时接通
A      SM0.5         // 在调整设置秒数时使数码管闪烁
ON     M0.4          // 在正常倒计时工作段接通
LD     M2.0          // 接通时秒数个位显示“2”
A      Q0.0          // 秒数个位七段数码管的公共端
LD     M3.4          // 接通时秒数十位显示“2”
A      Q0.1          // 秒数十位七段数码管的公共端
OLD
ALD
LD     M0.5          // 在调整设置分钟时接通
A      SM0.5         // 在调整设置分钟时使数码管闪烁
ON     M0.5          // 在正常倒计时工作段接通
LD     M5.0          // 接通时分钟个位显示“2”
A      Q0.2          // 分钟个位七段数码管的公共端
LD     M6.4          // 接通时分钟十位显示“2”
A      Q0.3          // 分钟十位七段数码管的公共端
OLD
ALD
OLD
LD     M7.2          // 接通时小时显示“22”
O      M8.4          // 接通时小时显示“12”
O      M9.6          // 接通时小时显示“2”
LD     M0.6          // 在调整设置小时时接通
A      SM0.5         // 在调整设置小时时使数码管闪烁
ON     M0.6          // 在正常倒计时工作段接通
ALD
A      Q0.4          // 小时个位七段数码管的公共端
OLD
LD     M7.1          // 接通时小时显示“23”
O      M7.2          // 接通时小时显示“22”
O      M7.3          // 接通时小时显示“21”
O      M7.4          // 接通时小时显示“20”
LD     M0.6          // 在调整设置小时时接通
A      SM0.5         // 在调整设置小时时使数码管闪烁
ON     M0.6          // 在正常倒计时工作段接通
```

```
ALD
A       Q0.5            // 小时十位七段数码管的公共端
OLD
LD      M0.7            // 在调整设置天数时接通
A       SM0.5           // 在调整设置天数时使数码管闪烁
ON      M0.7            // 在正常倒计时工作段接通
LD      M12.0           // 接通时天数个位显示“2”
A       Q0.6            // 天数个位七段数码管的公共端
LD      M14.0           // 接通时天数十位显示“2”
A       Q0.7            // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB    16#B6, QB1      // 将数字“2”送到输出端通过七段数码管显示
Network 18              // 七段数码管显示“1”的程序段
LD      M0.4            // 在调整设置秒数时接通
A       SM0.5           // 在调整设置秒数时使数码管闪烁
ON      M0.4            // 在正常倒计时工作段接通
LD      M2.1            // 接通时秒数个位显示“1”
A       Q0.0            // 秒数个位七段数码管的公共端
LD      M3.5            // 接通时秒数十位显示“1”
A       Q0.1            // 秒数十位七段数码管的公共端
OLD
ALD
LD      M0.5            // 在调整设置分钟时接通
A       SM0.5           // 在调整设置分钟时使数码管闪烁
ON      M0.5            // 在正常倒计时工作段接通
LD      M5.1            // 接通时分钟个位显示“1”
A       Q0.2            // 分钟个位七段数码管的公共端
LD      M6.5            // 接通时分钟十位显示“1”
A       Q0.3            // 分钟十位七段数码管的公共端
OLD
ALD
OLD
LD      M7.3            // 接通时小时显示“21”
```

```
O          M8.5              // 接通时小时显示“11”
O          M9.7              // 接通时小时显示“1”
LD         M0.6              // 在调整设置小时时接通
A          SM0.5             // 在调整设置小时时使数码管闪烁
ON         M0.6              // 在正常倒计时工作段接通
ALD
A          Q0.4              // 小时个位七段数码管的公共端
OLD
LD         M0.6              // 在调整设置小时时接通
A          SM0.5             // 在调整设置小时时使数码管闪烁
ON         M0.6              // 在正常倒计时工作段接通
AW>        MW7, 16           // 当 MW7 中数据大于 16 时，小时的十位显示 1
AD<        MD7, 32768        // 当 MD7 中数据小于 32768 时，小时的十位显示 1
A          Q0.5              // 小时十位七段数码管的公共端
OLD
LD         M0.7              // 在调整设置天数时接通
A          SM0.5             // 在调整设置天数时使数码管闪烁
ON         M0.7              // 在正常倒计时工作段接通
LD         M12.1             // 接通时天数个位显示“1”
A          Q0.6              // 天数个位七段数码管的公共端
LD         M14.1             // 接通时天数十位显示“1”
A          Q0.7              // 天数十位七段数码管的公共端
OLD
ALD
OLD
MOVB       16#0C, QB1        // 将数字“1”送到输出端通过七段数码管显示
Network 19                   // 七段数码管显示“0”的程序段
LD         M0.4              // 在调整设置秒数时接通
A          SM0.5             // 在调整设置秒数时使数码管闪烁
ON         M0.4              // 在正常倒计时工作段接通
LD         M1.0              // 接通时秒数个位显示“0”
A          Q0.0              // 秒数个位七段数码管的公共端
LD         M3.0              // 接通时秒数十位显示“0”
A          Q0.1              // 秒数十位七段数码管的公共端
OLD
```

```
ALD
LD      M0.5          // 在调整设置分钟时接通
A       SM0.5         // 在调整设置分钟时使数码管闪烁
ON      M0.5          // 在正常倒计时工作段接通
LD      M4.0          // 接通时分钟个位显示“0”
A       Q0.2          // 分钟个位七段数码管的公共端
LD      M6.0          // 接通时分钟十位显示“0”
A       Q0.3          // 分钟十位七段数码管的公共端
OLD
ALD
OLD
LD      M0.6          // 在调整设置小时时接通
A       SM0.5         // 在调整设置小时时使数码管闪烁
ON      M0.6          // 在正常倒计时工作段接通
LD      M7.0          // 接通时小时显示“0”
O       M7.4          // 接通时小时显示“20”
O       M8.6          // 接通时小时显示“10”
A       Q0.4          // 小时个位七段数码管的公共端
LD      M8.7          // 此点闭合小时显示“9”时，小时的十位显示 0
O       M7.0          // 此点闭合小时显示“0”时，小时的十位显示 0
OW<=    MW9, 128      // 当小时数小于 10 时，小时的十位显示 0
A       Q0.5          // 小时十位七段数码管的公共端
OLD
ALD
OLD
LD      M11.0         // 接通时天数个位显示“0”
A       Q0.6          // 天数个位七段数码管的公共端
LD      M13.0         // 接通时天数十位显示“0”
A       Q0.7          // 天数十位七段数码管的公共端
OLD
LD      M0.7          // 在调整设置天数时接通
A       SM0.5         // 在调整设置天数时使数码管闪烁
ON      M0.7          // 在正常倒计时工作段接通
ALD
OLD
```

```
MOVB    16#7E, QB1          // 将数字“0”送到输出端通过七段数码管显示
Network 20                  // 各七段数码管公共端轮流导通程序段
LD      SM0.0
LPS
AN      T34
TON     T33, 5
LRD
A       T33
TON     T34, 5
LRD
A       T34                 // 每只七段数码管公共端导通 100ms
RLB     QB0, 1              // 按 100ms 的时间使各公共端轮流导通
LPP
A       SM0.1
MOVB    1, QB0              // 开机时将数据 1 送入公共端以备轮流
```

例二、圆形停车库汽车存取控制程序

1. 控制要求

圆形停车库共有六个泊位，如图 5-3 所示。钥匙开关 QS1～QS6 分别为 6 个泊位的选择开关，SQ1～SQ6 为汽车在位限位开关，车库只设一个进出口并设门区信号 SQ7。

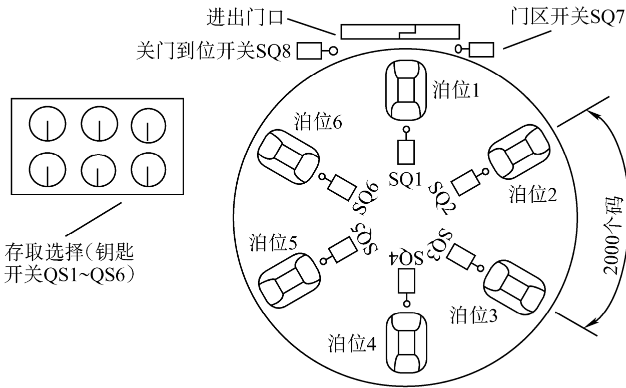


图 5-3　圆形停车库汽车存取转盘示意图

（1）当控制系统开始运行工作时，PLC 登记当前处在进出口位置的泊位号。

（2）钥匙开关是个两挡开关，当钥匙插进锁头往左拧是存车，往右拧是取车，

不用时开关处在空挡。当有存取车信号时，PLC 记录此位号并判断是存还是取，然后圆盘按照离请求泊位号最近的方向转动。转盘转动到进出口位置停止，转盘停止后打开出口门，10s 后关门，结束一次存取，等待下一个信号。

（3）转盘转动距离由旋转编码器计算，利用 PLC 的高速计数器功能处理此信号，按计算距离到达门口并进入门区才可以停靠，假设泊位间距的编码器码数为 2000 个。

（4）在处理某一个请求信号过程中，其他存取请求信号均无效，处理完当前信号并记录此信号，才可以接收下一个请求信号。用七段数码管显示处在门口的泊位号。

（5）只给出开关门信号，不考虑门系统控制，接到关门到位信号后转盘才能转动。

2. 程序设计

（1）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配，通过本题应了解和掌握高速计数器在运动物体计算位移方面的应用，本题还应用到转换指令、比较指令、数学运算指令、传送指令等。

（2）汽车存取 PLC 控制接线图如图 5-4 所示。手动调整系统在这里不考虑。

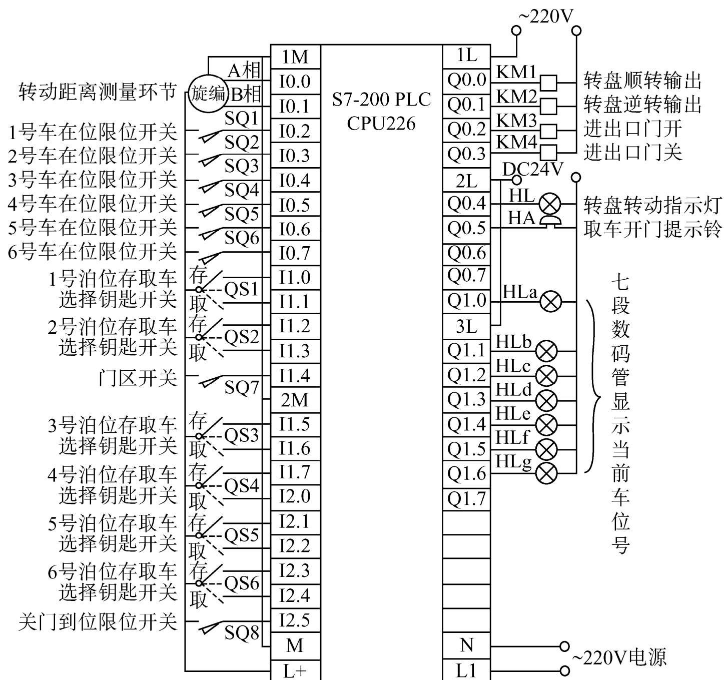


图 5-4　圆形停车库汽车存取 PLC 控制接线图

（3）汽车存取 PLC 控制语句表程序及注释如下：

```
主程序                                程序注释
Network 1                            // 当前门口泊位号显示与存储程序段
LD      M2.0
O       M2.1
O       SM0.1
ITB     VW2, VB20                    // 将字转变成字节用来七段数码管显示
SEG     VB20, QB1                    // 七段数码管显示当前门口泊位号
MOVW    VW2, VW30                    // 存储当前门口泊位号以利下次登记比较
Network 2                            // 呼叫登记排队程序段
LD      I2.5                         // 在进出口门确已关好的前提下
LPS
LD      I1.1                         // 1 号泊位取车钥匙开关信号
A       I0.2                         // 在 1 号泊位有车的前提下
LD      I1.0                         // 1 号泊位存车钥匙开关信号
AN      I0.2                         // 在 1 号泊位没有车的前提下
OLD
AN      M5.0
ALD
MOVW    1, VW2                       // 将数字 1 送到存储器中
S       M5.0, 1                      // 每一过程只处理一个请求信号，其他信号无效
LRD
LD      I1.3                         // 2 号泊位取车钥匙开关信号
A       I0.3                         // 在 2 号泊位有车的前提下
LD      I1.2                         // 2 号泊位存车钥匙开关信号
AN      I0.3                         // 在 2 号泊位没有车的前提下
OLD
AN      M5.0
ALD
MOVW    2, VW2                       // 将数字 2 送到存储器中
S       M5.0, 1                      // 每一过程只处理一个请求信号，其他信号无效
LRD
LD      I1.6                         // 3 号泊位取车钥匙开关信号
A       I0.4                         // 在 3 号泊位有车的前提下
LD      I1.5                         // 3 号泊位存车钥匙开关信号
```

```
AN      I0.4        // 在 3 号泊位没有车的前提下
OLD
AN      M5.0
ALD
MOVW    3, VW2      // 将数字 3 送到存储器中
S       M5.0, 1     // 每一过程只处理一个请求信号，其他信号无效
LRD
LD      I2.0        // 4 号泊位取车钥匙开关信号
A       I0.5        // 在 4 号泊位有车的前提下
LD      I1.7        // 4 号泊位存车钥匙开关信号
AN      I0.5        // 在 4 号泊位没有车的前提下
OLD
AN      M5.0
ALD
MOVW    4, VW2      // 将数字 4 送到存储器中
S       M5.0, 1     // 每一过程只处理一个请求信号，其他信号无效
LRD
LD      I2.2        // 5 号泊位取车钥匙开关信号
A       I0.6        // 在 5 号泊位有车的前提下
LD      I2.1        // 5 号泊位存车钥匙开关信号
AN      I0.6        // 在 5 号泊位没有车的前提下
OLD
AN      M5.0
ALD
MOVW    5, VW2      // 将数字 5 送到存储器中
S       M5.0, 1     // 每一过程只处理一个请求信号，其他信号无效
LPP
LD      I2.4        // 6 号泊位取车钥匙开关信号
A       I0.7        // 在 6 号泊位有车的前提下
LD      I2.3        // 6 号泊位存车钥匙开关信号
AN      I0.7        // 在 6 号泊位没有车的前提下
OLD
AN      M5.0
ALD
MOVW    6, VW2      // 将数字 6 送到存储器中
```

```
S        M5.0, 1          // 每一过程只处理一个请求信号，其他信号无效
Network 3                 // 门口泊位号与请求信号比较程序段
LD       M5.0
LPS
AW>      VW30, VW2        // 本次的请求信号与此刻处在门口的泊位号比较
=        M0.0             // 门口泊位号大于请求信号时 M0.0 继电器得电
LRD
AW=      VW30, VW2        // 门口泊位号等于请求信号时 M0.1 继电器得电
=        M0.1
LRD
AW<      VW30, VW2        // 门口泊位号小于请求信号时 M0.2 继电器得电
=        M0.2
LRD
A        M0.0
EU
MOVW     VW30, VW4        // 如门口泊位号大于请求信号，则两者相减结果肯
-I       VW2, VW4         // 定大于零，将结果放入 VW4 存储器中
LRD
A        M0.1             // 如门口泊位号等于请求信号
A        I1.4             // 泊位也正好在门区范围内
AN       Q0.0             // 此刻转盘没有顺转
AN       Q0.1             // 此刻转盘也没有逆转
=        Q0.2             // 以上条件都具备，那就直接开门
LPP
A        M0.2
MOVW     VW30, VW6        // 如门口泊位号小于请求信号，则两者相减结果
+I       +6, VW6          // 肯定小于零，只好先将泊位号加上 6 然后再减
MOVW     VW6, VW4         // 去请求信号，将结果放入 VW4 存储器中
-I       VW2, VW4
Network 4                 // 决定顺转或逆转及转动站数计算程序段
LD       M5.0
LPS
AW>=     VW4, 3           // 当请求信号与门口泊位号之差大于等于 3 时
=        M1.0             // 此继电器闭合具备逆转条件
MOVW     +6, VW8          // 用 6 减去门口泊位号与请求信号的差放入
```

```
–I      VW4, VW8       //  VW8 存储器中，这就是需转动的站数
AW=     VW10, VW8      // 当高速计数器计算的站数与 VW8 中的相等时
=       M2.0           // 此继电器闭合使转盘停转
LRD
AW<     VW4, 3         // 当请求信号与门口泊位号之差小于 3 时
=       M1.1           // 此继电器闭合具备顺转条件
AW=     VW10, VW4      // 当高速计数器计算的站数与 VW4 中的相等时
=       M2.1           // 此继电器闭合使转盘停转
LRD
AN      Q0.2           // 在没有开门的前提下
LPS
A       M1.0           // 逆转条件具备的前提下
AN      M1.1           // 应没有顺转条件
AN      Q0.0           // 没有顺转
AN      M2.0           // 门口泊位号与请求信号有距离的前提下
=       Q0.1           // 转盘开始逆转，到门口时 M2.0 得电，Q0.1 断电
LPP                    // 停转等待开门
A       M1.1           // 顺转条件具备的前提下
AN      M1.0           // 应没有逆转条件
AN      Q0.1           // 没有逆转
AN      M2.1           // 门口泊位号与请求信号有距离的前提下
=       Q0.0           // 转盘开始顺转，到门口时 M2.1 得电，Q0.0 断电
LRD                    // 停转等待开门
LD      Q0.2           // 开门后要有一个 10s 延迟
O       M0.3
AN      Q0.3
ALD
=       M0.3
TON     T37, 100       // 延迟时间为 10s
AW<     T37, 20
=       Q0.5           // 在开门过程的前 2s 有指示灯点亮
LRD
LD      T37
O       Q0.3
ALD
```

```
AN        I2.5              // 关门到位后此点断开
=         Q0.3              // 10s 后就开始关门
LRD
LD        Q0.0
O         Q0.1
ALD
A         SM0.5
=         Q0.4              // 不管是顺转或逆转只要转动就有指示灯闪亮
LRD
A         Q0.1              // 当逆转时高速计数器将按下列数据计算停车
LPS
AD=       HC0, -2000        // 当高速计数器计数到此数时就逆转过 1 个泊位
EU
MOVW      1, VW10           // 立即把 1 送到 VW10 中去与请求信号比较
LRD
AD=       HC0, -4000        // 当高速计数器计数到此数时就逆转过 2 个泊位
EU
MOVW      2, VW10           // 立即把 2 送到 VW10 中去与请求信号比较
LPP
AD=       HC0, -6000        // 当高速计数器计数到此数时就逆转过 3 个泊位
EU
MOVW      3, VW10           // 立即把 3 送到 VW10 中去与请求信号比较
LPP
A         Q0.0              // 当顺转时高速计数器将按下列数据计算停车
LPS
AD=       HC0, 2000         // 当高速计数器计数到此数时就顺转过 1 个泊位
EU
MOVW      1, VW10           // 立即把 1 送到 VW10 中去与请求信号比较
LPP
AD=       HC0, 4000         // 当高速计数器计数到此数时就顺转过 2 个泊位
EU
MOVW      2, VW10           // 立即把 2 送到 VW10 中去与请求信号比较
Network 5                   // 1 个过程结束后需完成的指令段
LD        I2.5              // 当门关好后，高速计数器就要初始化一次
EU
```

```
O        SM0.1            // 每次开机，高速计数器也要初始化一次
CALL     SBR0             // 调用高速计数器初始化的子程序
R        M5.0, 1          // 1 个过程结束后，此继电器就要复位
MOVW     0, VW10          // 1 个过程结束后要给 VW10 存储器清零
子程序                          程序注释
Network 1                 // 高速计数器初始化程序段
LD       SM0.0
MOVB     16#FC, SMB37     // 给高速计数器定状态位
HDEF     0, 9             // 给高速计数器定工作模式
HSC      0                // 起动 0 号高速计数器
```

例三、T68 镗床的 PLC 改造程序

1. 控制要求

主轴可以正反转，并分高低速运行，还可以实现反接制动。反接制动是靠速度继电器配合反向接法接触器来共同完成的，主轴正向起动时，按 SB2 按钮，然后 KM3 闭合，之后 KM1 闭合，再之后 KM4 闭合，主轴电动机此时以正转低速运行。要变高速时，只要扳动主轴变速手柄，将此手柄置于高速位置，这时限位开关 SQ7 被压，它接通了通电延时继电器 KT，经过延时后，KM4 断开，KM5 闭合，主轴电动机就变为高速了。无论何时主轴变高速，都要先经低速后再到高速，制动时，只要按 SB1 停止按钮，速度继电器 KS 的常开触点 KS 就会配合反向接触器 KM2，把速度立刻降下来，如果先起动了反向，整个过程与正向相同，只是起制动作用的是 KM1 了。

如果在主轴工作过程中需变速，这时不用按停止按钮 SB1。只要将主轴变速操作盘的操作手柄拉出，使行程开关 SQ3 不再受压，SQ5 也不再受压，使 KM3，KT 线圈断电释放，KM1（或 KM2）也随之断电，之后反接制动回路又能形成使 KM2（或 KM1），KM4 线圈立即通电吸合。电动机 1M 在低速状态下串电阻反接制动，当制动结束后，便可转动变速操纵盘进行变速。变速后，将手柄推回原位，使 SQ3、SQ5 的触点恢复原来状态，使 KM3、KM1（或 KM2）、KM4 的线圈相继通电吸合。电动机按原来的转向起动，而以新选定的转速运转。变速时，若手柄出现卡住问题，这时 SQ5 配合 KS 的常闭触点，周期性地使 KM1、KM4 线圈相继通电吸合，直至齿轮啮合后，卡住问题消失，方可推回操纵手柄，变速冲动才算结束。

如果主轴在进给过程中希望变速，这时只要拉开进给变速操作手柄即可，不过这时与之相关联的行程开关是 SQ4 与 SQ6。动作过程与主轴转动是相同的。切记，这两个变速过程是靠两个操作手柄来完成的，各负其责。如果这两个操作手

柄同时都被扳动，那么这时与之相对应的 SQ1 与 SQ2 就都断开了，控制电路全部断电，换句话说，这两个手柄不能同时动作。

快速进给电动机 2M 的控制。这里又有一个操作手柄，名称叫快速进给操作手柄，将它向里推，压合行程开关 SQ9，使 KM6 线圈通电吸合，2M 正向起动。松开手柄，快速进给停止。SQ9 复位，使 KM6 失电，反之，向外拉手柄时。压合 SQ8，使 KM7 线圈吸合，电动机反向起动。

总之，镗床上共有 2 台电动机，分别为 1M 和 2M。1M 负责主轴的旋转和进给，2M 是快速进给电动机，有 3 个操作手柄，有 2 个与 1M 有关，1 个与 2M 有关，名称是主轴变速操作手柄。主轴进给变速操作手柄。快速进给操作手柄。有 9 个行程开关，SQ1 与 SQ2 负责两个变速手柄的联锁，不可同时动作。SQ3、SQ4、SQ5、SQ6 都与变速手柄有关，SQ7 与主轴变高速时有关。SQ8、SQ9 快速进给的两个方向。有 7 个接触器，其中 KM1—主轴正转；KM2—主轴反转；KM3—起动时闭合，制动时断开用来串电阻；KM4—主轴低速；KM5—主轴高速；KM6—快速正向进给；KM7—快速反向进给；KT—时间继电器，低向高转换延时。本次改造是以 1994 年中国劳动出版社出版，劳动部培训司组织编写的《维修电工生产实习》（第 2 版）为脚本，T68 镗床电气原理图如图 5-5 所示，结构示意图如图 5-6 所示。

2. 程序设计

（1）PLC 改造 T68 镗床电气控制系统接线图如图 5-7 所示。

（2）PLC 控制程序语句表及注释如下：

```
主程序                              程序注释
Network 1
LD      I2.1               // 主轴电动机过载时此点断开
A       I1.2               // 主轴箱与主轴不能同时为快速
A       I2.0               // 停止按钮
LPS
LD      I0.1               // 主轴正向起动按钮
O       M0.1
ALD
AN      M0.2
=       M0.1               // 主轴正向运行继电器
LRD
LD      I0.2               // 主轴反向起动按钮
O       M0.2
ALD
```

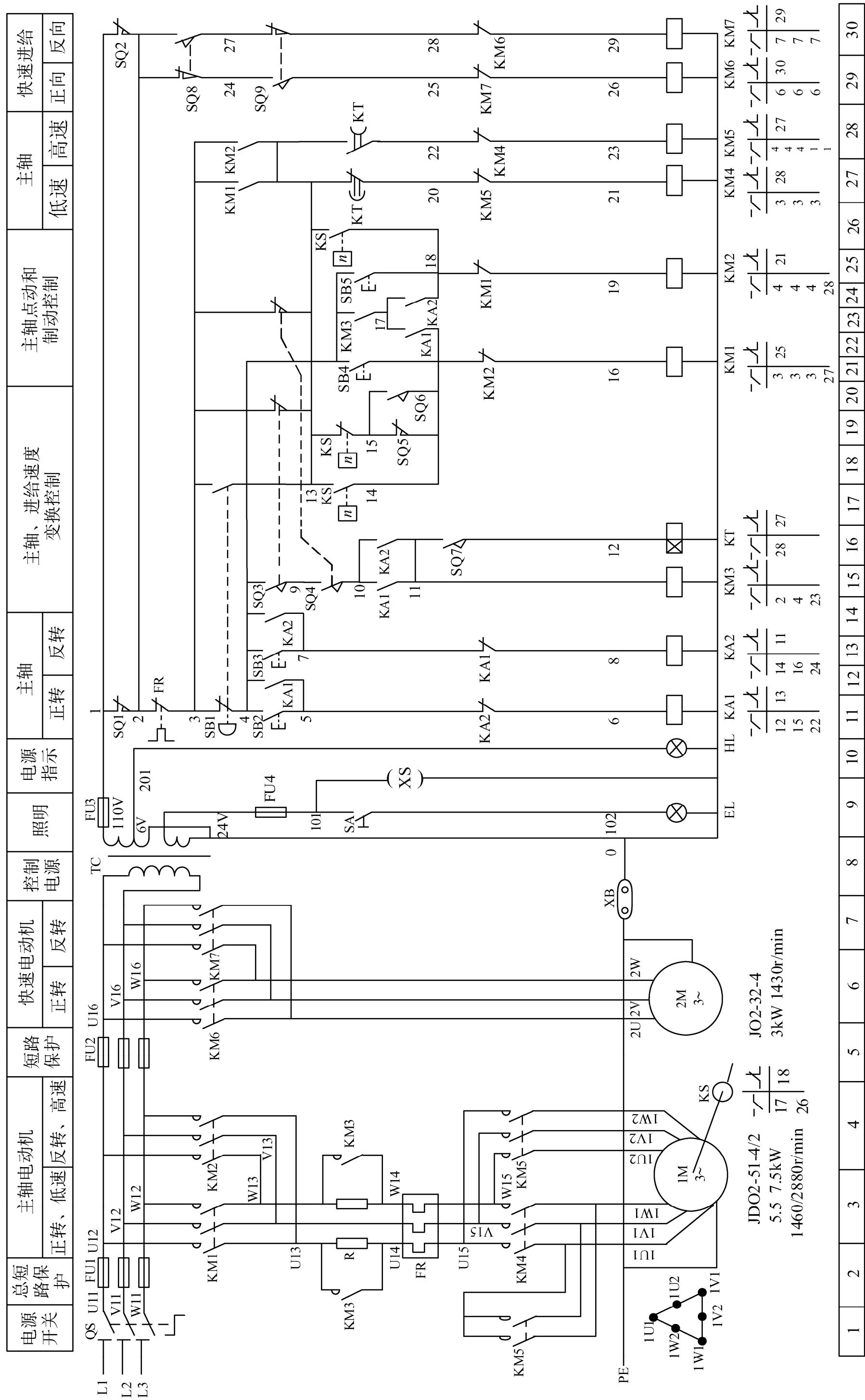


图 5-5　T68 镗床电气原理图

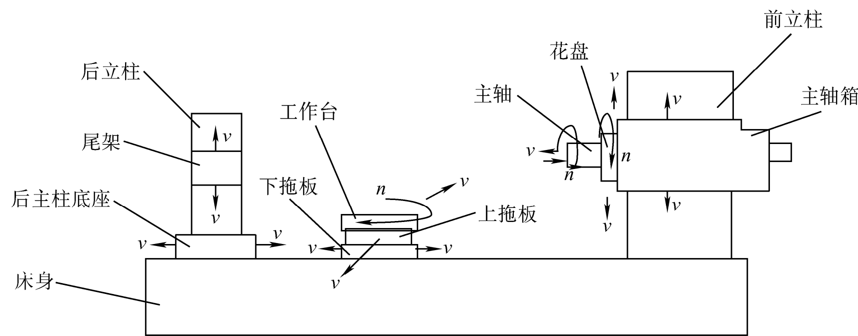


图 5-6　T68 镗床结构示意图

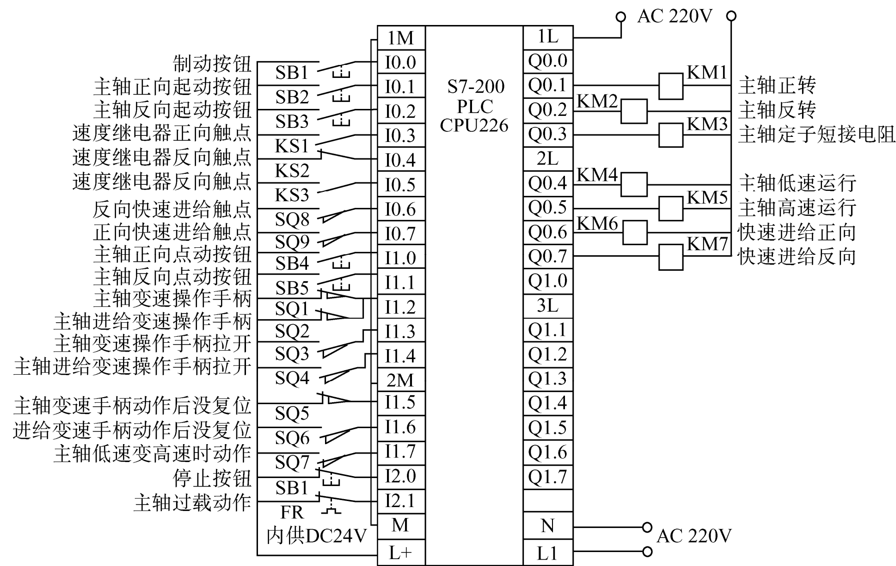


图 5-7　PLC 改造 T68 镗床的电气控制系统接线图

```
AN      M0.1
=       M0.2            // 主轴反向运行继电器
LPP
LD      M0.1
O       M0.2
A       I1.3            // 主轴变速操作中此点断开
A       I1.4            // 主轴进给变速操作中此点断开
ALD
```

```
=       Q0.3            // 主轴不带电阻起动
A       I1.7            // 主轴低速变高速时此点闭合
TON     T37, 30         // 主轴快、慢速转换延时
Network 2
LD      I1.5            // 如主轴变速手柄动作后没复位时此点闭合
O       I1.6            // 如主轴进给变速手柄动作后没复位时此点闭合
LDN     I1.3            // 主轴变速手柄动作时此点闭合
ON      I1.4            // 主轴进给变速手柄动作时此点闭合
ALD
A       I0.4            // 主轴低速时此点闭合
LD      M0.1            // 主轴正常正向运行时此点闭合
A       Q0.3
O       I1.0            // 主轴正向点动按钮
OLD
AN      Q0.2            // 主轴反向互锁
A       I2.0            // 停止按钮
LD      I0.0            // 制动按钮
O       Q0.1
A       I0.3            // 速度继电器触点，此刻应闭合
OLD
A       I1.2            // 两种变速操作手柄都没动作时此点闭合
A       I2.1            // 主轴电动机没过载时此点闭合
=       Q0.1            // 主轴电动机正向运行接触器
Network 3
LD      M0.2
A       Q0.3
O       I1.1            // 主轴电动机反向运行点动
AN      Q0.1            // 主轴正向互锁
A       I2.0            // 停止按钮
LD      I0.0            // 制动按钮
O       Q0.2
A       I0.5            // 速度继电器触点，此刻应闭合
OLD
A       I1.2            // 两种变速操作手柄都没动作时此点闭合
A       I2.1            // 主轴电动机没过载时此点闭合
```

```
=       Q0.2            // 主轴电动机反向运行接触器
Network 4
LD      I1.2
LPS
LD      Q0.1
O       Q0.2
ALD
LPS
AN      T37             // 主轴低速转高速延时
A       I2.1
AN      Q0.5
=       Q0.4            // 主轴低速运行接触器
LPP
A       T37
A       I2.1
AN      Q0.4
=       Q0.5            // 主轴高速运行接触器
LPP
LPS
A       I0.7            // 正向快速进给触点
AN      I0.6
AN      Q0.7            // 正反向快速进给互锁
=       Q0.6            // 正向快速进给接触器
LPP
A       I0.6            // 反向快速进给触点
AN      I0.7
AN      Q0.6            // 正反向快速进给互锁
=       Q0.7            // 反向快速进给接触器
```

例四、X62W 万能铣床的 PLC 改造程序

1. 控制要求

X62W 万能铣床共有三台电动机，可以实现主轴电动机 1M 的正反转，由接触器 KM1 加换相开关 SA4 共同实现。主轴的制动由接触器 KM2 与继电器 KS 配合实现，主轴的变速冲动由接触器 KM2 与行程开关 SQ7 共同实现。工作台进给电动机 2M 可实现六个方向的移动，即上、下、前、后、左、右。进给电动机 2M

只有在 1M 得电动作后才能动作，虽然有六个方向可移动，但某一时刻只进行其中一个方向的移动，定向是通过两个机械手柄，它们之间是联锁控制的，如果使用这个手柄，则那个手柄必须处在中间停止位置，否则这个手柄无法扳动。具体分工：向前、向下、向右由 KM3 实现；向后、向上、向左由 KM4 实现。此外六个方向不进行铣切加工时，工作台能快速移动，方法是在主轴电动机起动后，将进给手柄扳到所需位置，工作台按照选定的速度和方向的常速进给，再按下快速进给按钮 SB5 或 SB6 使接触器 KM5 得电吸合，接通牵引电磁铁 YA，不需要快速移动时，松开 SB5 或 SB6 按钮就可以了。进给变速瞬时冲动的实现是通过瞬时接通接触器 KM3 来实现的，这时两个手柄都必须处在中间位置上，也就是停止位置上，因为要使用 SQ1～SQ4 的常闭触点。

此外工作台还可以做圆周运动，这时两个手柄也是应处在中间（停止）位置上，是通过扳动组合开关 SA1 从而接通 KM3 来实现的，回转运动只能是单向的，没有反转，且在此时，如误操作扳动了操作手柄，则电动机立即停止。本次改造是以 1994 年中国劳动出版社出版，劳动部培训司组织编写的《维修电工生产实习》（第二版）为脚本。原电气控制系统原理图如图 5-8 所示，X62W 万能铣床的加工示意图如图 5-9 所示。

2. 程序设计

（1）PLC 改造 X62W 万能铣床电气控制系统接线图如图 5-10 所示。

（2）PLC 控制程序语句表及注释如下：

```
主程序                      程序注释
LD      I1.6        // 主轴电动机如过载时此点断开
LPS
LD      I0.0        // 起动主轴电动机可以在两个地方
O       Q0.0
ALD
AN      I0.1        // 如按下制动按钮此点断开，主轴电动机停转
AN      Q0.1        // 制动过程中不能起动电动机
AN      I0.3        // 如主轴电动机需变速，应先使主轴电动机停止
=       Q0.0        // 控制主轴电动机运行的接触器
LRD
LD      I0.1        // 使主轴电动机制动运行的按钮
O       Q0.1
A       I0.2        // 速度继电器触点，当速度降下来后断开制动
AN      I0.3        // 如需变速冲动，断开制动回路
O       I0.3        // 需变速冲动时，相应的限位开关动作闭合
```

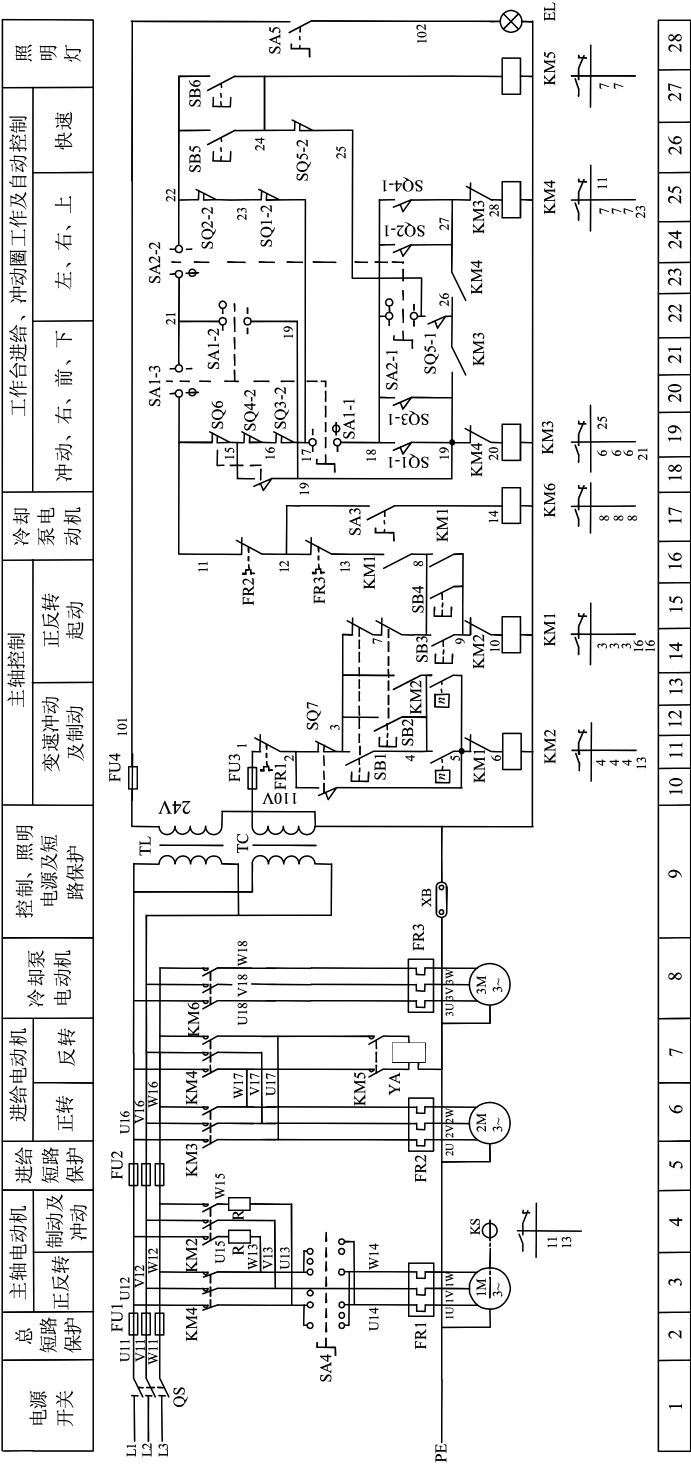


图 5-8 X62W 万能铣床电气原理图

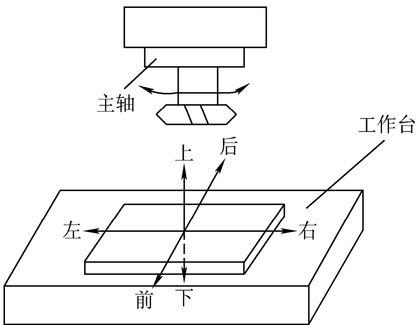


图 5-9　X62W 万能铣床加工示意图

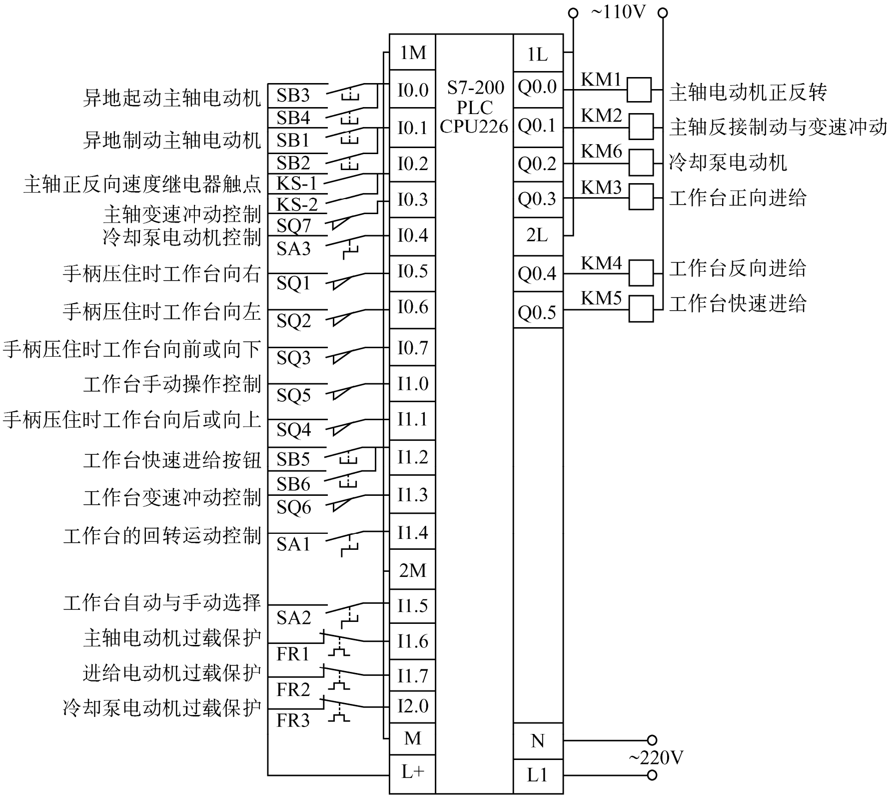


图 5-10　PLC 的控制系统接线图

```
ALD
AN      Q0.0        // 断开运行状态才能进行制动
=       Q0.1        // 控制主轴电动机制动运行的接触器
```

```
LPP
A       Q0.0        // 在主轴电动机起动运行后才能有下面的动作
LPS
A       I0.4        // 起动冷却泵电动机
AN      I0.3        // 变速冲动时将使冷却泵停止
AN      I0.1        // 制动时将使冷却泵停止
A       I2.0        // 如冷却泵电动机过载时此点断开
=       Q0.2        // 控制冷却泵电动机运行的接触器
LPP
AN      I1.5        // 自动与手动开关定在手动位置
A       I1.7        // 如进给电动机过载时此点断开
LPS
LD      I1.4        // 工作台回转运动控制
A       I1.3        // 工作台变速冲动时此点闭合
LDN     I1.4        // 工作台回转运动控制
AN      I1.3        // 工作台不进行变速冲动时此点闭合
OLD
AN      I0.5        // 工作台在进行回转运动或变速冲动时断开右行
AN      I0.6        // 工作台在进行回转运动或变速冲动时断开左行
AN      I0.7        // 工作台在进行回转运动或变速冲动时断开下及前行
AN      I1.1        // 工作台在进行回转运动或变速冲动时断开上及后行
LD      I0.7        // 工作台在前行及下行时此点闭合
AN      I0.5        // 工作台在前行及下行时不能有右行
AN      I0.6        // 工作台在前行及下行时不能有左行
AN      I1.1        // 工作台在前行及下行时不能有后行及上行
LD      I0.5        // 工作台在右行时此点闭合
AN      I0.6        // 工作台在右行时不能有左行
AN      I0.7        // 工作台在右行时不能有前行及下行
AN      I1.1        // 工作台在右行时不能有后行及上行
OLD
AN      I1.3        // 工作台在进行进给时不能有变速及冲动
A       I1.4        // 回转运动控制在闭合位置
OLD
ALD
AN      Q0.4        // 在进行右、前、下进给时不能有左、后、上操作
```

```
=     Q0.3      // 控制右、前、下方向进给的接触器
LPP
LPS
LD    I1.1      // 工作台在后行及上行时此点闭合
AN    I0.5      // 工作台在后行及上行时不能有右行
AN    I0.6      // 工作台在后行及上行时不能有左行
AN    I0.7      // 工作台在后行及上行时不能有前行及下行
LD    I0.6      // 工作台在左行时此点闭合
AN    I0.5      // 工作台在左行时不能有右行
AN    I0.7      // 工作台在左行时不能有前行及下行
AN    I1.1      // 工作台在左行时不能有后行及上行
OLD
ALD
A     I1.4      // 回转运动控制在闭合位置
AN    Q0.3      // 在进行左、后、上进给时不能有右、前、下操作
=     Q0.4      // 控制左、后、上方向进给的接触器
LPP
LD    I0.5      // 在各个方向进给时都可以进行快速进给
O     I0.6
O     I0.7
O     I1.1
ALD
A     I1.2      // 快速进给控制按钮
=     Q0.5      // 快速进给控制接触器
```

例五、两台 PLC 主从式通信程序设计

1. 控制要求

某食品杀菌设备要求有一个热水储备罐，把水放进里面加热到所需温度，然后再把热水送到杀菌罐中进行食品杀菌处理。现因储备罐距电气控制柜较远，为节省资源将储备罐上的水泵控制线、水位信号线、电磁阀控制线等都接到附近另一个控制柜中，将两个控制柜中的 PLC 用通信线连起来进行主从式通信，这样既保证能完成整体杀菌过程控制，又节省了很多资源。储备罐储水过程示意图如图 5-11 所示。

现在我们将该设备的主控制柜称为主站，将储备罐附近的控制柜称为从站。当我们在主控制柜那里按动开始杀菌处理的起动按钮时，从站那里接到此信号打

开给水阀（阀 1），同时起动水泵，将水放进储备罐中，当水位到达设定水位时，水位计中的触点闭合，送给从站 PLC 一个信号，PLC 接到此信号后停止进水转入下一个工序给水加热，打开蒸汽阀（阀 2），蒸汽进来使罐内温度逐渐升高，热电阻作为感温元件传递温度信号，经温度变送器将信号传给温度控制仪，当水温到达设定值时控制仪上的触点闭合，将信号送给从站 PLC，PLC 接到信号后关闭蒸汽阀，然后打开注入阀（阀 3），将热水注入到处理罐中，假设 30s 后热水全部注入到处理罐中，此时关闭注入阀，进入下一个工序。因后面的生产过程与通信无关，所以程序就编写到此。

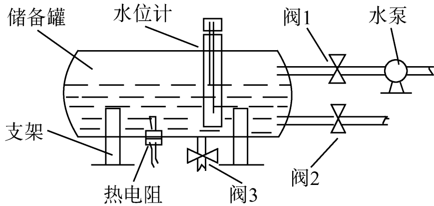


图 5-11　储备罐储水过程示意图

2. 编程思路

这是两台 PLC 主从式通信的例子，通过这个例子我们应了解主从式通信的编程特点就是作为主站的 PLC 担子很重，因为它是主角，所以读写程序都在主站中编写，从站中有时甚至没有程序，本题的特点就是从站中没有程序。编程者一定要清楚主站读从站数据的区域在哪里，当那里的数据有变化时，主站都能知道并加以利用，同时主站要把哪些内容写给从站，写到从站的哪个位置也应是很清楚的。建立了这些初始化程序，剩下的就是实现具体控制要求了。

3. 程序设计

（1）两台 PLC 的控制系统接线图如图 5-12 所示，图中主站的输出信号因与本题无关所以略去，主站的输入信号 SB1 为整个控制过程的起动按钮，SB2 为停止

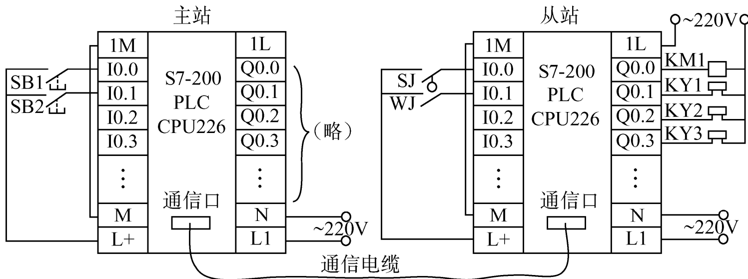


图 5-12　两台 PLC 的控制系统接线图

按钮；从站的输入信号 SJ 为水位计水位信号，WJ 为水温温度到达信号，从站的输出信号 KM1 为水泵接触器线圈，KY1 为给水阀（阀 1），KY2 为蒸汽阀（阀 2），KY3 为注入阀（阀 3）。

主从式通信方式的特点就是经济实惠，接线简单，编程容易，在图中我们可以看到两台 PLC 之间用一根专用通信电缆通过各自的同号通信口就可连接起来，进行信号传递。

（2）PLC 控制程序语句表及注释如下：

```
主站程序                          // 程序注释
Network 1                        // 确定通信模式与通信口程序段
LD      SM0.1                    // 每当 PLC 从停止转为运行都要初始化一次
MOVB    2, SMB130                // 模式为主从式并通过通信口 1 进行通信
FILL    0, VW200, 10             // 把负责完成通信任务的所有寄存器清 0
Network 2                        // 主站读取从站数据的初始化程序段
LDN     SM0.1                    // PLC 从停止转为运行后的第二个周期开始闭合
AN      V200.6                   // 操作排队有效后此点断开
AN      V200.5                   // 通信程序编写有错误时此点断开
MOVB    3, VB201                 // 指定从站地址为 3
MOVD    &IB0, VD202              // 指定读取从站数据的区域位置
MOVB    1, VB206                 // 读取从站的数据长度为 1B
NETR    VB200, 1                 // 由 VB200 为首的数据表负责执行读指令
MOVB    VB207, MB2               // 将读进来的单字节数据放入 MB2 中
Network 3                        // 主站写给从站数据的初始化程序段
LDN     SM0.1                    // PLC 从停止转为运行后的第二个周期开始闭合
AN      V210.6                   // 操作排队有效后此点断开
AN      V210.5                   // 通信程序编写有错误时此点断开
MOVB    3, VB211                 // 指定从站地址为 3
MOVD    &QB0, VD212              // 指定写入从站数据的区域位置
MOVB    1, VB216                 // 写入从站的数据长度为 1B
MOVB    MB0, VB217               // 将主站的 MB0 中的数据写给从站
NETW    VB210, 1                 // 由 VB210 为首的数据表负责执行写指令
Network 4                        // 具体完成控制要求的程序段
LDN     I0.1                     // 停止按钮，处在主站的输入口
LPS
LD      I0.0                     // 起动按钮，处在主站的输入口
O       M1.0
```

```
ALD
=       M1.0        // 保证按动起动按钮后才能有各种动作的继电器
LRD
LD      M1.0
O       M0.0
ALD
AN      M2.0        // 水位到达后此点断开，输入信号由从站传来
=       M0.0        // 按下起动按钮后如水位未到即起动水泵进水
LRD
LPS
A       M0.0
TON     T37, 5      // 水泵起动后延迟 0.5s 打开给水阀才可进水
LPP
A       T37
=       M0.1        // 给水阀继电器
LRD
LD      M2.0        // 水位到达后此点闭合，信号来自从站输入端
O       M0.2
ALD
AN      M0.0        // 水泵停止时此点闭合
AN      M2.1        // 水温未到时此点闭合
A       M1.0        // 在按动起动按钮后才可进行工作
=       M0.2        // 起动蒸汽阀继电器
LRD
LD      M2.0        // 水位到达此点闭合
A       M2.1        // 温度到达此点闭合
O       M0.3
ALD
AN      M0.2        // 蒸汽阀不工作时此点闭合
A       M1.0        // 在按动起动按钮后才可进行工作
AN      T38
=       M0.3        // 起动注入阀继电器
LPP
A       M0.3
TON     T38, 30     // 经 3s 后所有水都已放出，关闭注入阀
```

例六、用 TD200 监控邮包配送的程序设计

1. 本题控制要求

某邮包配送机构，有一个总站，两个分站。由总站向两个分站批送，每次从总站装 20 件邮包，送往两分站各 10 件，然后返回继续装车。第一个分站的邮包数到 40 件时就不要了。第二个分站到 60 件时也不要了，空车返回。

要求：用 TD200 文本显示器实现控制与监视。按要求需 3 个计数开关，3 个位置开关，起动按钮及停止按钮各 1 个，加在一起这 8 个开关全部由 TD200 上的 8 个按钮来模拟代替。运行状态、站位号、装卸邮包数量都由显示器窗口显示出来。

2. 连接与设置

TD200 的外形图如图 5-13 所示，在 S7-200 PLC 通电前就应把 TD200 与 S7-200 PLC 连接好，如图 5-14 所示，可以用 TD/CPU 电缆方便地将两者通过通信端口连接起来，然后给 S7-200 PLC 通上电源，两者就都通电了。TD200 的显示窗口点亮并显示本 TD200 的型号与版本，接下来又显示 CPU 的状态，这时我们可以通过计算机（已经通过通信口与 S7-200 PLC 建立好通信关系）上的 STEP 7-Micro/WIN 编程软件进行设置（组态）。

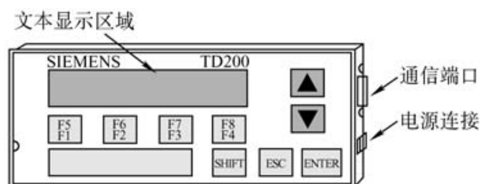


图 5-13　TD200 外形图

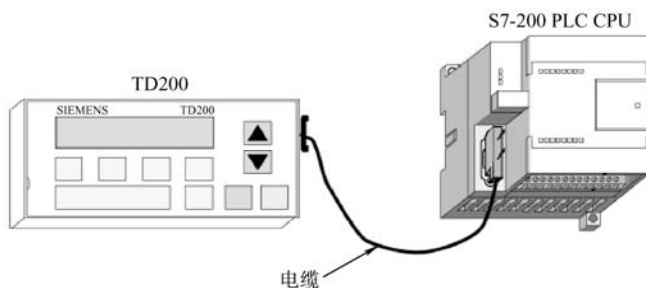


图 5-14　TD200 与 PLC 连接示意图

关于设置（组态），我们可参见第四章中的图 4-50 至图 4-63，只有经过这些设置后，才可以进行编程调试。本题需设置的内容如下：①将 8 个按钮设置成本题所需的起停按钮、计数开关、位置开关，并为其找到相对应的变量寄存器（V）的位。②工作状态设置（TD200 为我们显示的内容）：送出、返回、正在总站装包、

正在第一分站卸包、正在第二分站卸包。③计数设置（变量设置）：正在装第 *n* 件邮包（其中 *n* 为变量）、正在卸第 *n* 件邮包（其中 *n* 为变量）。本题的 TD200 与变量寄存器的对应地址首地址选择的是 VB200，这样选择后的对应关系如图 5-15 所示。图中指给我们这个画面打开的路径是指令树→符号表→向导→TD_SYM_200，从图 5-15 中可看出为 8 个按钮所配置的 S7-200 PLC 中的 V 寄存器的地址位以及需 TD200 显示的 7 条工作状态（也就是上面的②与③）的 V 寄存器的地址位。编程时把这些地址位都对应编进去就可以了。

	符号	地址	注释
1	S_F4	V259.7	键盘按键"SHIFT+F4"已按下标志（瞬动触点）
2	F4	V257.3	键盘按键"F4"已按下标志（瞬动触点）
3	S_F3	V259.6	键盘按键"SHIFT+F3"已按下标志（瞬动触点）
4	F3	V257.2	键盘按键"F3"已按下标志（瞬动触点）
5	S_F2	V259.5	键盘按键"SHIFT+F2"已按下标志（瞬动触点）
6	F2	V257.1	键盘按键"F2"已按下标志（瞬动触点）
7	S_F1	V259.4	键盘按键"SHIFT+F1"已按下标志（瞬动触点）
8	F1	V257.0	键盘按键"F1"已按下标志（瞬动触点）
9	TD_CurScreen_200	VB263	TD 200 显示的当前屏幕（其配置起始于 VB200）。如无屏幕显示则设置为 16#FF。
10	TD_Left_Arrow_Key_200	V256.4	'左箭头'键按下时置位
11	TD_Right_Arrow_Key_200	V256.3	'右箭头'键按下时置位
12	TD_Enter_200	V256.2	'ENTER'键按下时置位
13	TD_Down_Arrow_Key_200	V256.1	'下箭头'键按下时置位
14	TD_Up_Arrow_Key_200	V256.0	'上箭头'键按下时置位
15	TD_Reset_200	V245.0	此位置位会使 TD 200 从 VB200 重读其配置信息。
16	Alarm0_6	V246.1	报警使能位 6
17	Alarm0_5	V246.2	报警使能位 5
18	Alarm0_4	V246.3	报警使能位 4
19	Alarm0_3	V246.4	报警使能位 3
20	Alarm0_2	V246.5	报警使能位 2
21	Alarm0_1	V246.6	报警使能位 1
22	Alarm0_0	V246.7	报警使能位 0

图 5-15　与 TD200 相对应的变量寄存器的地址

3. 程序设计

（1）S7-200 PLC 与 TD200 文本显示器之间通信电缆连接示意图及 PLC 对外接线图如图 5-16 所示。

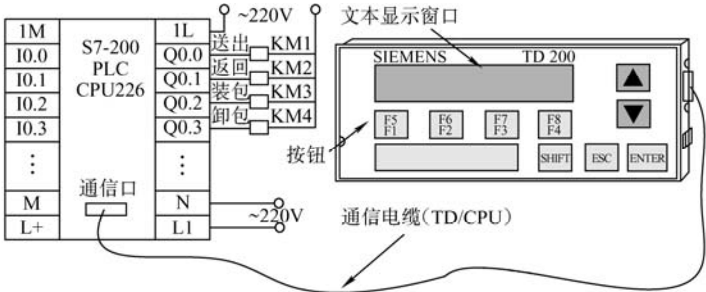
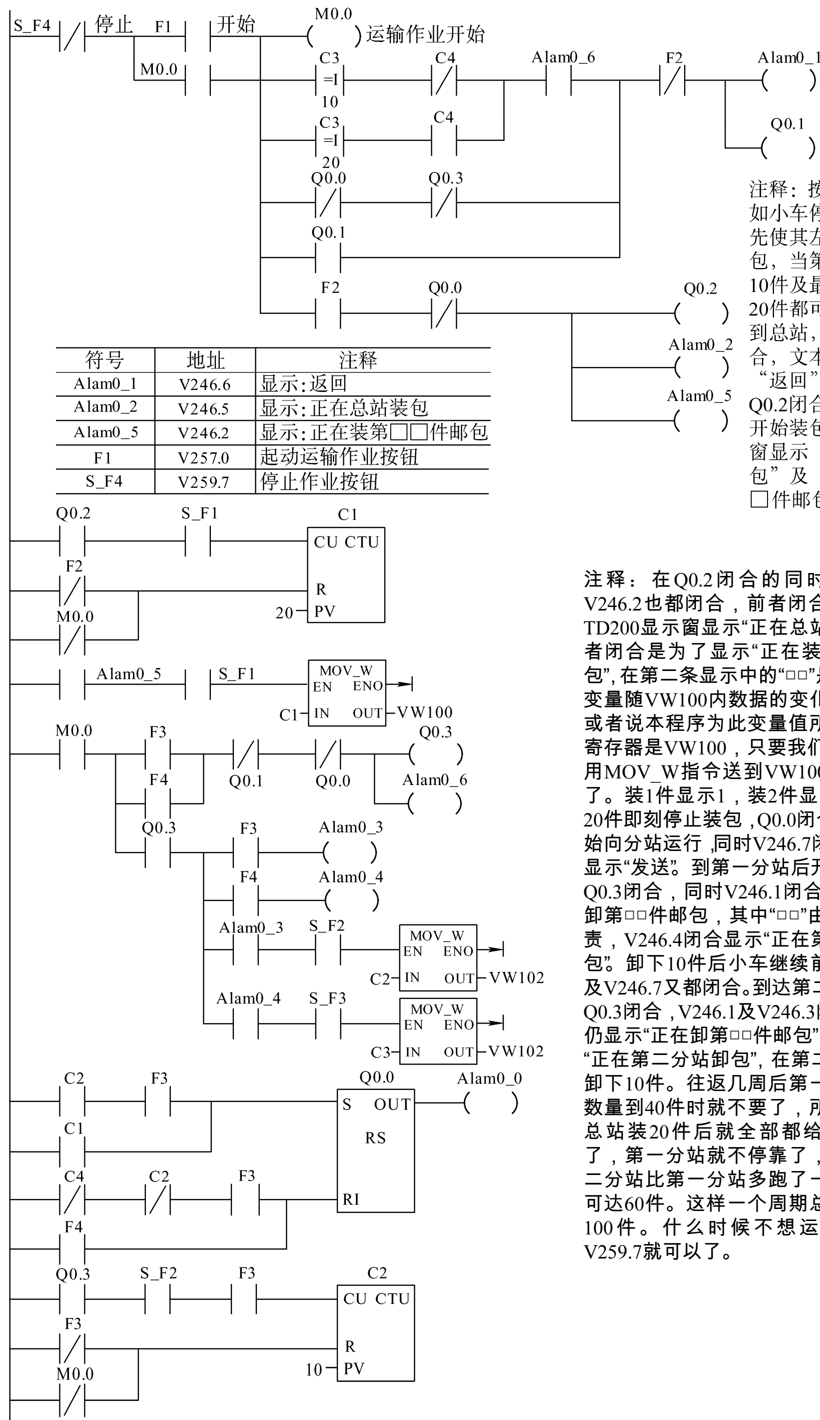


图 5-16　PLC 与 TD200 连接示意及 PLC 对外接线图

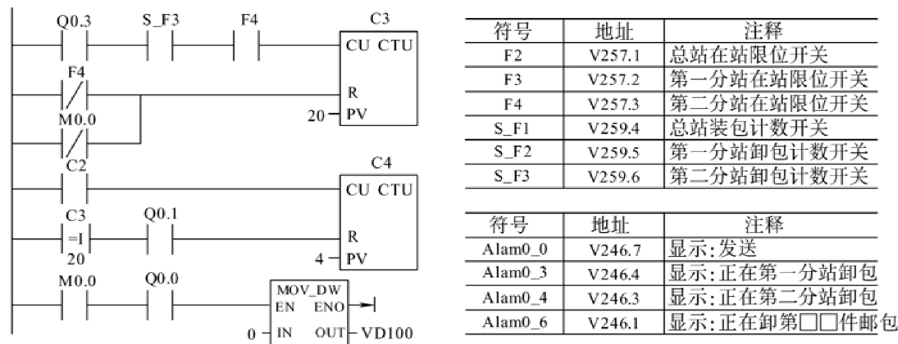
（2）控制程序的梯形图及注释如下：



符号	地址	注释
Alam0_1	V246.6	显示:返回
Alam0_2	V246.5	显示:正在总站装包
Alam0_5	V246.2	显示:正在装第□□件邮包
F1	V257.0	起动运输作业按钮
S_F4	V259.7	停止作业按钮

注释：按起动按钮后如小车停在任意位置先使其左行到总站装包，当第二分站卸完10件及最后一次卸完20件都可使小车左行到总站，此时Q0.1闭合，文本显示窗显示“返回”。到总站后Q0.2闭合起动装包机开始装包，文本显示窗显示“正在总站装包”及“正在装第□□件邮包”。

注释：在Q0.2闭合的同时V246.5及V246.2也都闭合，前者闭合是为了在TD200显示窗显示“正在总站装包”，后者闭合是为了显示“正在装第□□件邮包”，在第二条显示中的“□□”是变量，此变量随VW100内数据的变化而变化，或者说本程序为此变量值所定的变量寄存器是VW100，只要我们把C1的值用MOV_W指令送到VW100中就可以了。装1件显示1，装2件显示2，装满20件即刻停止装包，Q0.0闭合，小车开始向分站运行，同时V246.7闭合显示窗显示“发送”。到第一分站后开始卸包，Q0.3闭合，同时V246.1闭合显示“正在卸第□□件邮包，其中“□□”由VW102负责，V246.4闭合显示“正在第一分站卸包”。卸下10件后小车继续前行，Q0.0及V246.7又都闭合。到达第二分站变为Q0.3闭合，V246.1及V246.3闭合，前者仍显示“正在卸第□□件邮包”，后者显示“正在第二分站卸包”，在第二分站也是卸下10件。往返几周后第一分站邮包数量到40件时就不要了，所以这时从总站装20件后就全部都给第二分站了，第一分站就不停靠了，或者说第二分站比第一分站多跑了一趟，总数可达60件。这样一个周期总站要发包100件。什么时候不想运输了按下V259.7就可以了。



符号	地址	注释
F2	V257.1	总站在站限位开关
F3	V257.2	第一分站在站限位开关
F4	V257.3	第二分站在站限位开关
S_F1	V259.4	总站装包计数开关
S_F2	V259.5	第一分站卸包计数开关
S_F3	V259.6	第二分站卸包计数开关

符号	地址	注释
Alam0_0	V246.7	显示:发送
Alam0_3	V246.4	显示:正在第一分站卸包
Alam0_4	V246.3	显示:正在第二分站卸包
Alam0_6	V246.1	显示:正在卸第□□件邮包

例七、基于 PID 的食品罐头杀菌温度控制程序设计

1. 任务描述

肉类罐头食品的杀菌温度一般是 121℃，到达此温度后就开始恒温运行。温度低于此值达不到灭菌效果，而高于此值又会出现焦煳变色影响质量，如采用电磁阀作为蒸汽进汽阀，因其不能控制开度，待测温电阻感测到设定值时，罐内的整体温度也许已超过设定值，控制温度的曲线就会出现如图 5-17 所示的超标振荡现象。为了避免这种现象使曲线既快速平滑又不会超标，那就要采取 PID 控制，把电磁阀换成开度可控的电动阀，这样通过我们设置就可形成实际值（过程变量）与设定值的温差越大电动阀的开度也越大，反之温差越小开度也越小，所形成的曲线如图 5-18 所示。现设定最高温度为 150℃，它的 80%正好是 121℃，待温度升到 80%时既是给定值（SPn），这样电动阀的开度就会随着温差变小而逐步变小，较平滑地接近恒温温度。

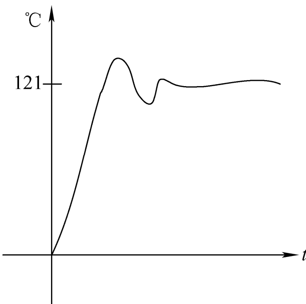


图 5-17　不带 PID 控制的电磁阀的升温曲线

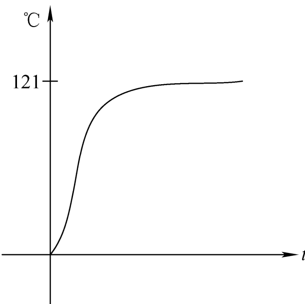


图 5-18　PID 控制的电动阀的升温曲线

采用 PID 控制功能完成本题的程序设计，在硬件上除了 S7-200 PLC 主机之外，我们还需增加一块 EM235 模拟量扩展模块、3 线式热电阻一个。在软件上采用 PID 控制只是控制升温段及恒温段，恒温段后面的杀菌处理过程本题不考虑。

具体控制过程是这样的，在121℃（150℃的80%）之前全量程开启电动阀，经过PID计算，过程变量当前值越是接近给定值电动阀的开度就越小，温度的变化范围是150℃的0%～100%，是一个单极性信号，控制参数为：K_c=0.4、T_s=2s、T_i=10min、T_d=5min、M_n=0.8、输出信号的类型为0～10V电压输出型。

食品罐头杀菌罐的示意图如图5-19所示，电动阀为本题主要控制对象。

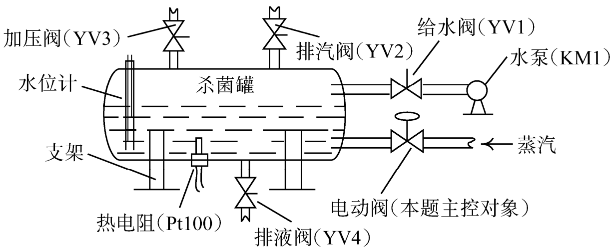


图5-19　食品罐头杀菌罐示意图

2. 控制过程说明

参考图5-19，将罐头食品放进杀菌罐中按下起动按钮，这时水泵（KM1）应起动、给水阀（YV1）打开，向杀菌罐中注水，待水位到达设定值时水位计中的触点（SL1）将会闭合，断开给水阀及水泵，这时电动阀开始工作，向罐中放进蒸汽。按照控制参数的要求经过PID运算决定电动阀的开度，待加热到设定温度值时，关闭电动阀开始进入恒温段，在恒温过程中如温度又低于设定值则再打开电动阀，开度由PID运算决定，原则是温差越小开度就越小，如温度超过设定值则就将排汽阀（YV2）定时打开，使温度降到设定值，控温曲线达到如图5-18所示的走向。恒温需要30min，恒温结束后即进入冷却段，在此不考虑后面的控制。

3. PLC对外接线说明

PLC控制系统的对外接线如图5-20所示，使用S7-200 PLC实现模拟量控制必须加模拟量扩展模块，在此选用的是EM235型的，它有四路模拟量输入口及一

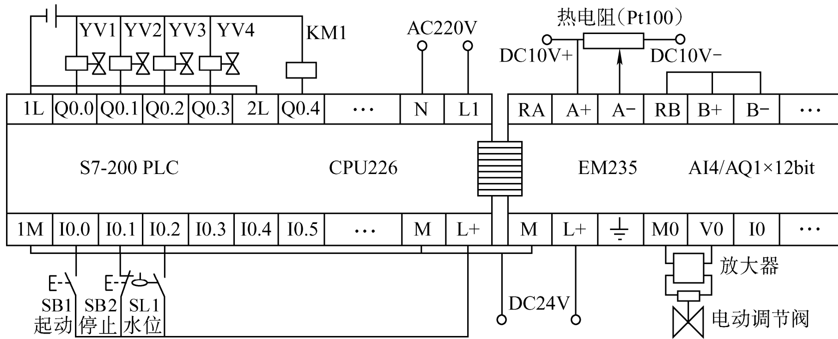


图5-20　PLC控制系统对外接线图

路模拟量输出口。输入侧的模拟量输入端口可直接接铂电阻而省去变送器环节，空闲的输入端口一定要用导线短接以免干扰信号入侵。取 12 位分辨率，满量程的模拟量（DC10V）对应的转换后的数据为 4 000。输出侧的模拟量输出端口可输出电压信号也可输出电流信号，负载需要什么就取什么。模拟量扩展模块的电源是 DC24V，这个电源一定要外接而不可就近接 PLC 本身输出的 DC24V 电源，但两者一定要共地。

4. 程序设计

子程序　　注释

M0.0
MOV_R EN ENO IN OUT：0.8 → VD304　　将温度设定值送入指定寄存器
MOV_R EN ENO IN OUT：0.4 → VD312　　将回路增益值送入指定寄存器
MOV_R EN ENO IN OUT：10.0 → VD320　　将积分时间常数送入指定寄存器
MOV_R EN ENO IN OUT：5.0 → VD324　　将微分时间常数送入指定寄存器

主程序　　注释

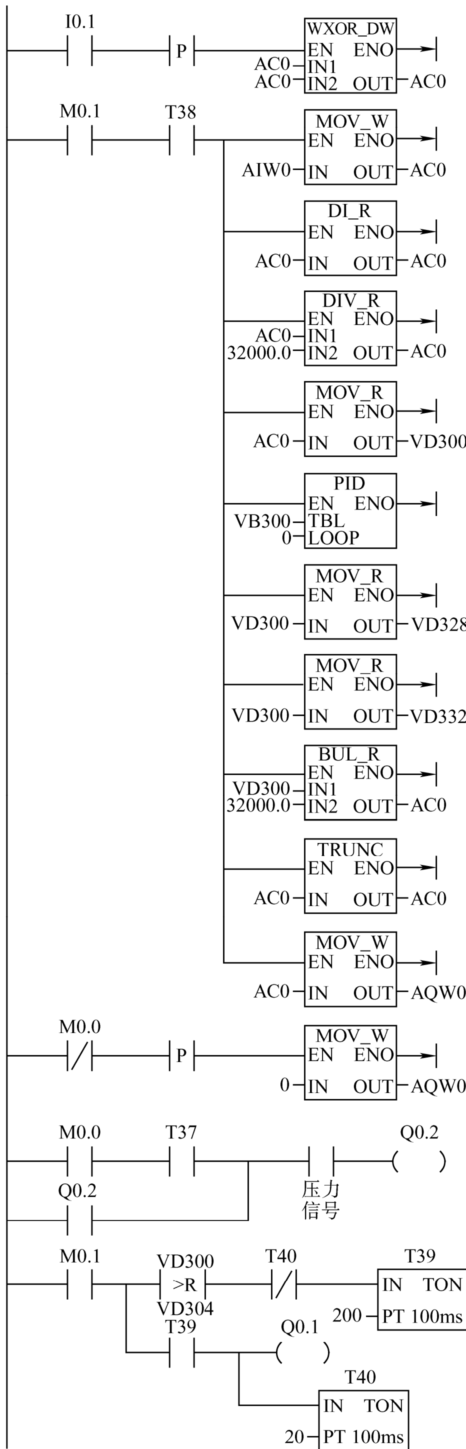
I0.0　P　SBR_0 EN　　罐内水位到位后立即激活温度PID控制子程序

I0.1　I0.0　M0.0
M0.0
I0.2　M0.1　Q0.0
Q0.4
I0.2　T37　M0.1
M0.1　VD300 ==R VD304　M0.3 S 1
M0.3　T37 IN TON　18000 PT 100ms
T37　M0.3 R 1

按起动按钮I0.0闭合使M0.0得电自保，从而Q0.0得电给水阀打开，同时Q0.4也得电使水泵工作，这时开始向罐内注水。等到水位高度达到要求时水位计触点I0.2闭合，使Q0.0及Q0.4失电，停止给水，同时激活温度PID控制程序使控温电动阀全程打开，罐内温度逐渐升高电动阀开度逐渐减小，等到当前值等于设定值时，M0.3得电置位开始进入恒温段，恒温时间是30min。

M0.1　M0.2　T38 IN TON　20 PT 100ms
T38　M0.2

温度值采样周期为1次/2s



将累加器 AC0 清零，准备接收过程变量（模拟量）的数字信号

将过程变量（模拟量）的数字信号送到累加器AC0中

将送进来的整数转换为实数

将已转换的实数被32000除，即转为标准化值

将此标准化值存入VD300中

将此标准化值存入PID参数表的VB300中

将计算后的输出值作为积分前项值存放到指定累加器中

将计算后的输出值作为最近一次PID运算的过程变量值存放到指定累加器中

将被控制量与实数32000相乘结果放到AC0中

将运算结果的实数格式转换成整数

再将运算结果直接以整数格式送到模拟量输出端口

恒温结束后应有一个反压过程，压力值由仪表设定

在恒温段罐内温度如高于设定值将定时打开排汽阀降温。具体方法是只要温度高于设定值隔20s排汽2s

例八、基于 USS 协议库的 PLC 与变频器的通信

1. USS 通信协议简介

USS通信协议专用于S7-200 PLC和西门子公司的Micro Master变频器之间的通信，这一系列的变频器都支持 USS 协议作为通信链路。通信介质由 S7-200 PLC 的通信接口和变频器内置的 RS-485 通信接口及屏蔽双绞线组成，采用半双工通信方式，数字化的信息传递，提高了系统的自动化水平及运行的可靠性，解决了模拟信号传输所引起的干扰及漂移问题，最远可达 1000m。可有效地减少电缆的数量，从而大大减少开发和工程费用，并极大地降低客户的起动和维护成本，通信效率较高，可达 187.5kbit/s。一台 S7-200 PLC CPU 最多可以监控 31 台变频器。接线量少，占用 PLC 的 I/O 点数少，传送的信息量大，只要把 PLC 的程序编写准确，就可随时控制变频器的起/停、改变运行频率及读/写参数，实现多台变频器的联动和同步控制。这是一种廉价的、编程容易的、使用方便的通信方式。

使用 USS 通信协议，用户程序可以通过子程序调用的方式实现 PLC 与变频器之间的通信，编程的工作量很小。在使用 USS 协议之前，需要在 STEP 7 编程软件中先安装“STEP 7-Micro/WIN V32 指令库”，几秒钟即可安装好。USS 协议指令在此指令库的文件夹中，指令库提供 8 条指令来支持 USS 协议，调用一条 USS 指令时，将会自动增加一个或多个相关的子程序。调用方法是打开 STEP 7 编程软件，在指令树的“\指令\库\USS Protocol”文件夹中，将会出现用于 USS 协议通信的指令，用它们来控制变频器和读写变频器参数。用户不需要关注这些子程序的内部结构，只要将有关指令的外部参数设置好，直接在用户程序中调用它们即可。

2. USS 协议指令

USS 协议指令主要包括 USS_INIT、USS_CTRL、USS_RPM、USS_WPM 四种。

① USS_INIT 指令　USS_INIT 指令如图 5-21 所示，用于初始化或改变 USS 的通信参数，只激活一次即可，也就是只需一个扫描周期就可以了。在执行其他 USS 协议指令之前，必须先执行 USS_INIT 指令，且没有错误返回。指令执行完后，完成位（Done）立即置位，然后才能继续执行下一条指令。

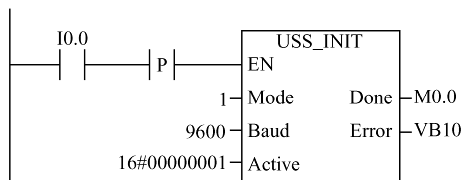


图 5-21　USS_INIT 指令

当 EN 端输入有效时，每一次扫描都会执行指令，这是不可以的。应通过一个边沿触发指令或特殊继电器 SM0.1，使此端只在一个扫描周期有效，激活指令就可以了。一旦 USS 协议已起动，如想改变初始化参数，必须通过执行一个新的

USS_INIT 指令以终止旧的 USS 协议。

“Mode”用于选择通信协议，字节型数据。如数据为 1，是将端口 0 分配给 USS 协议和允许该协议；如数据为 0，是将端口 0 分配给 PPI，并禁止 USS 协议。也就是说如没其他附加条件，USS 协议只能通过 PLC 的 0 号通信端口通信。

“Baud”用于设定波特率，单位为 bit/s，字型数据。可选 1 200、2 400、4 800、9 600、19 200、38 400、57 600 或 115 200bit/s。一定要与变频器参数所定的波特率一致。例如：MM440 变频器的 P2010 参数就是设定串行接口波特率的。

“Active”用于指示哪一个变频器是激活的，双字型数据。Active 共 32 位（第 0～31 位），例如第 0 位为 1 时，则表示激活 0 号变频器；第 0 位为 0 则不激活它。

“Done”用于指示指令执行情况，布尔型数据。指令执行完成后，此位为“1”。

“Error”用于生成一个字节，字节型数据。这一字节包含指令执行情况的信息。

② USS_CTRL 指令　USS_CRTL 指令如图 5-22 所示，是变频器控制指令，用于控制 Micro Master 变频器。USS_CTRL 指令将用户命令放在一个通信缓冲区内，如果由“Drive”指定的变频器被 USS_INIT 指令中的“Active”参数选中，缓冲区中的命令将被发送到该变频器。每个变频器只应有一个 USS_CTRL 指令，使用 USS_CTRL 指令的变频器应确保已被激活。

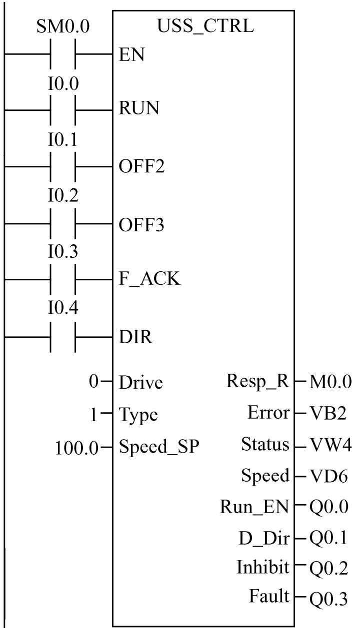


图 5-22　USS_CTRL 指令

EN 位必须接通，以起动 USS_CTRL 指令。一般情况下，这个指令总是处于允许执行状态，所以在图 5-22 中，此端用了一个 SM0.0（常 ON）触点。

“RUN”指示变频器是接通（1）或是断开（0）。当 RUN 位是接通时，Micro Master 变频器收到一个命令，以便开始以规定的速度和方向运行。为了使变频器运行，必须具备以下条件：在 USS_INIT 中将变频器激活；减速停止（OFF2）端和急停（OFF3）端必须为断态（OFF）；输出端 Fault 和 Inhibit 必须为 0；当 RUN 位断开时，则发送 Micro Master 变频器一个命令，电动机或减速停止或立即停止。

“OFF2”用来使 Micro Master 变频器减速到停止。“OFF3”用来使 Micro Master 变频器快速停止。

“F_ACK”（故障确认）用来确认一个故障。当 F_ACK 从低变高时，变频器清除故障，Fault 位恢复为 0。当出现故障时，“Error”端会有相应输出，且“Fault”

端变为 1，把故障处理完，给这个点一个“ON”信号，变频器才可恢复正常。

“DIR”（方向）用来设置变频器的运行方向（0-逆时针方向，1-顺时针方向）。

“Drive”（变频器地址）是 USS_CTRL 命令指定的 Micro Master 变频器地址，有效地址为 0～31。

“Type”是变频器的类型，3 系列或更早的系列为 0，4 系列的为 1。

“Speed_SP”速度设定点，是用全速度的百分比表示的速度设定值（–200.0%～200.0%）。该值为负时变频器反方向旋转。例：40Hz 就写 80（40/50x100%=80%）。

“Resp_R”（收到响应）确认从变频器来的响应。对所有激活的变频器轮询最新的变频器状态信息。每当 PLC 给变频器命令及信息，变频器收到后再返回一个响应，Reap_R 位便接通一个扫描周期，并更新以下所有的数值：

“Error”是一个错误状态字节，它包含与变频器通信请求的最新结果。

“Status”是由变频器返回的状态字的原始值。

“Speed”是变频器返回的用全速度百分比表示的变频器速度（–200.0%～200.0%）。相当于一个反馈信号，反映变频器实际运行速度。

“Run_EN”（RUN 允许）是用于指示变频器的运行状态，正在运行（1）或已停止（0）。

“D_Dir”用于指示变频器的旋转方向（0-逆时针方向，1-顺时针方向）。

“Inhibit”指示变频器上的禁止位的状态（0-不禁止，1-被禁止）。要清除禁止位，Fault 位必须为 0，RUN、OFF2 及 OFF3 输入位也必须为 0 状态。

“Fault”指示故障位的状态（0-无故障，1-故障）。发生故障时，变频器将提供故障代码（参阅变频器使用手册），在“Error”所指字节里，图 5-22 中 VB2 里。要清除 Fault 位，需找出故障原因并消除故障，然后接通 F_ACK 位。

表 5-1 给出 USS_CTRL 子程序各端子的操作数和数据类型。

表 5-1 USS_CTRL 子程序的操作数和数据类型

输入/输出	操 作 数	数据类型
RUN	I, Q, M, S, SM, T, C, V, L, 功率流	布尔数
OFF2	I, Q, M, S, SM, T, C, V, L, 功率流	布尔数
OFF3	I, Q, M, S, SM, T, C, V, L, 功率流	布尔数
F_ACK	I, Q, M, S, SM, T, C, V, L, 功率流	布尔数
DIR	I, Q, M, S, SM, T, C, V, L, 功率流	布尔数
Drive	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD	字节
Speed_SP	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD, 常数	实数
Resp_R	I, Q, M, S, SM, T, C, V, L	布尔数
Error	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	字节

（续）

输入/输出	操　作　数	数据类型
Status	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	字
Speed	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	实数
Run_EN	I, Q, M, S, SM, T, C, V, L	布尔数
D_Dir	I, Q, M, S, SM, T, C, V, L	布尔数
Inhibit	I, Q, M, S, SM, T, C, V, L	布尔数
Fault	I, Q, M, S, SM, T, C, V, L	布尔数

③ USS_RPM_W 指令　USS_RPM_W 指令格式如图 5-23 所示，用于读取变频器的无符号字，是 PLC 读取变频器参数的 3 条指令之一。当 Micro Master 变频器对接收的命令进行应答或返回一个出错状况时，则完成 USS_RPM_W 指令的处理。在该处理等待响应时，逻辑扫描仍继续进行。EN 位必须接通以起动发送请求，这个位应保持接通一直到 Done 位被置位才标志着整个处理结束。

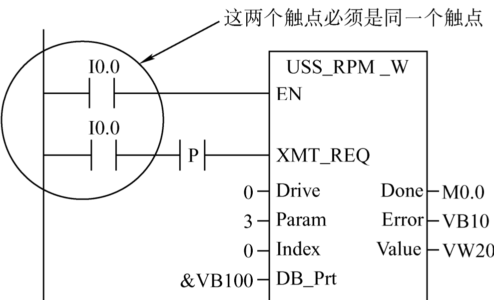


图 5-23　USS_RPM_W 指令

当 XMT_REQ 位输入接通时，每次扫描，USS_RPM 都发送请求到变频器，因此，XMT_REQ 的输入端必须与脉冲边沿检测指令相连接，保证每次 EN 输入端到来时，XMT_REQ 输入端只接通一个扫描周期，用来向变频器发出请求。

Drive 是变频器的地址，字节变量，USS_RPM 命令将被发送到这个地址，每个变频器的有效地址为 0～31。

字变量 Param 和 Index 分别是要读取的变频器参数的编号和参数的下标值。

必须将 16B 缓冲区的地址提供给 DB_Ptr 输入，USS_RPM 指令使用这个缓冲区以存储向变频器所发送命令的结果。

USS_RPM 指令完成时，Done 位输出接通，意味着所要的信息已读取过来，且 Error 位输出字节包含执行这个指令的结果。Value 是读取过来的参数值。

④ USS_WPM_W 指令　USS_WPM_W 指令格式如图 5-24 所示，用于写入变

频器的无符号字，是 PLC 写入变频器参数的 3 条指令之一。当 Micro Master 变频器对接收的命令进行应答或返回一个出错状况时，则完成 USS_WPM_W 指令的处理。在该处理等待响应时，逻辑扫描仍继续进行。EN 位必须接通以起动发送请求，这个位应保持接通一直到 Done 位被置位才标志着整个处理结束。

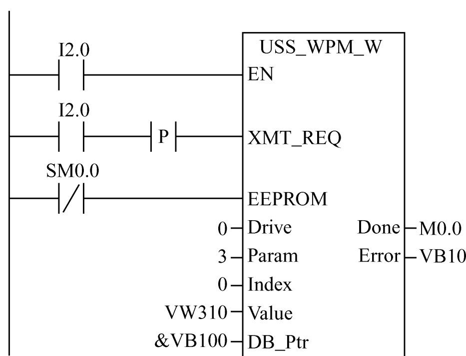


图 5-24 USS_WPM_W 指令

USS_WPM_W 指令中 EN、XMT_REQ、Drive、Param、Index、DB_Ptr、Done、Error 各位的作用与 USS_RPM_W 指令中的各位相同。Value 是要写入变频器的 RAM 的参数值，也可以写入变频器的 EEPROM。EEPROM 输入接通时，指令同时将参数写入变频器的 RAM 和 EEPROM，该输入断开时，只写入变频器的 RAM。

3. USS 的使用要求

USS 通信占用 0 号通信端口，在选择使用 USS 协议与驱动通信后，此端口不能够再用于其他用途，包括与 STEP 7-Micro/WIN 通信。只有通过执行另外一条 USS_INIT 指令，或将 CPU 的模式开关置于 STOP 位置，才能重新使端口 0 用于与 STEP 7-Micro/WIN 通信。PLC 与变频器的通信中断将使变频器停止工作。

USS 指令影响所有的与端口 0 的自由口通信相关的 SM 区。

USS 指令使用 14 个子程序、3 个中断程序和累加器 AC0～AC3。

USS 指令占用用户程序存储空间 2300～3600B。

USS 指令需要 400B 的 V 存储区。区域的起始地址由用户指定并保留给 USS 变量。作为用户不必关心留给 USS 指令的 V 存储区的内容，只记住别在他用。

有一些 USS 指令还要求 16B 的通信缓冲区。缓冲区的起始地址由用户指定。建议为每一条 USS 协议指令指定一个单独的缓冲区。仍不可再用于其他。

USS 指令不能用在中断程序中。

4. USS 的编程顺序

① 使用 USS__INIT 指令初始化变频器。定通信口、定波特率、定变频器地

址号。

② 使用 USS__CTRL 激活变频器。起动变频器、定变频器运行方向、定变频器减速停止方式、清除变频器故障、定运行速度、定与 USS__INIT 指令相同的变频器地址号。

③ 配置变频器参数，以便和 USS 指令中指定的波特率和地址相对应。

④ 连接 PLC 和变频器间的通信电缆。应特别注意变频器的内置式 RS-485 接口。

⑤ 程序输入时应注意，S7 系列的 USS 协议指令是成型的，在编程时不必理会 USS 的子程序和中断，只要在主程序中开启 USS 指令库就可以了。调用位置如图 5-25 所示。

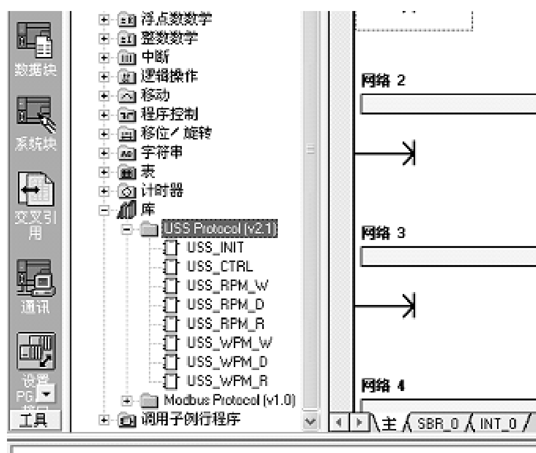


图 5-25　USS 协议指令在编程软件中的位置

5. 西门子公司 Micro Master 440 变频器

Micro Master 440 变频器是用于控制三相交流电动机速度的系列产品，Micro Master 440 变频器有多种规格，额定功率范围为 120W～200kW，或者可达 250kW，可供用户选择。

变频器具有缺省的工厂设置参数，可直接拖动电动机，实现电动机的变频调速运行。在设置了变频器的相关参数以后，既可以用作单独的驱动系统，也可以通过自身的并行口或串行口集成到自动化系统中。

在进行 Micro Master 440 变频器的通信电缆连接时，取下变频器的前盖板露出接线端子，如图 5-26 所示。将 RS-485 通信电缆的一端与变频器的 USS 专用端子相连，因为要与端子相连，所以不用带连接器，而 S7-200 PLC 那端的电缆头却一定要带连接器。

将变频器连接到 PLC 之前必须确认变频器已有以下的系统参数，可使用变频

器正面操作盒上的键盘设定参数，参数可按以下步骤设定：

① 将变频器复位到出厂时的设定值（或称缺省值）。使 P0010=30，P0970=1。然后按 P 键，再等几秒钟这个步骤就结束了，别人先前设置过的参数就都被抹掉了，免得影响正常设置。接下来要设置以下参数：

USS PZD 长度：P2012 [0]=2

USS PKW 长度：P2013[0] =127

② 能对变频器所有参数的读/写访问（专家模式）：P0003=3

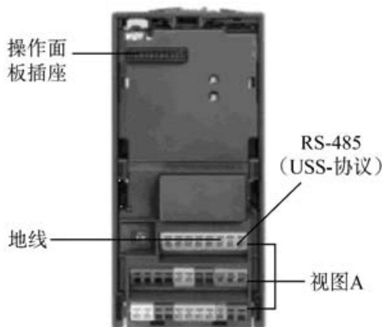


图 5-26 变频器接线端子的外形图

③ 检查所驱动的电动机设置：

P0304=额定电动机电压（V）

P0305=额定电动机电流（A）

P0307=额定功率（W）

P0310=额定电动机频率（Hz）

P0311=额定电动机速度（RPM）

这些设置因使用的电动机不同而不同。

要设置参数 P0304，P0305，P0307，P0310 和 P0311，必须先将参数 P0010 设为 1（快速调试模式）。当完成参数设置后，将参数 P0010 再设为 0。参数 P0304，P0305，P0307，P0310 和 P0311 只能在快速调试模式下修改。

④ 设定参数 P0700=5，设置为远程控制方式，即通过 RS-485 通信链路的 USS 通信。

⑤ 设定 RS-485 串行接口的波特率。

P2010[0]=4（2400bit/s）；=5（4800bit/s）；=6（9600bit/s 默认值）；=7（19200bit/s）。

⑥ 输入从站地址。每个变频器（最大 31）可经过总线运行。P2011[0]=0～31。

⑦ 设置基准频率。P2000=50Hz。

⑧ 设置 USS 规格化。P2009=0，禁止 USS 规格化；P2009=1，允许 USS 规格化。

⑨ EEPROM 存储器控制（任选）。当 P0971=0 断电时，丢失更改的参数设定值（包括 P0971）；当 P0971=1 断电时（默认值），仍保持更改的参数设定值。

⑩ 设定参数 P1000=5，即通过 RS-485（COM）通信链路的 USS 通信发送频率设定值。

其他参数可随时通过 PLC 程序写入变频器。

计算机、PLC、变频器和电动机之间的连接示意图如图 5-27 所示。

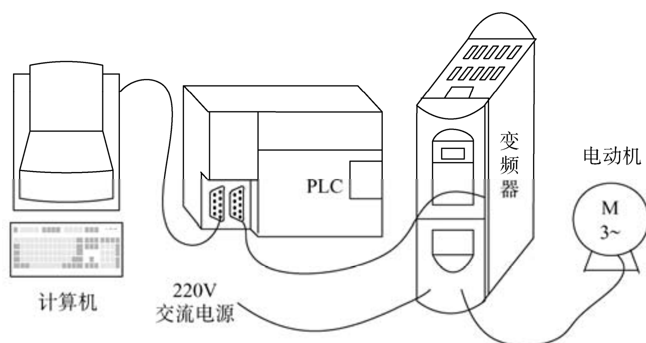


图 5-27　连接示意图

6. 本题控制要求

按下参数输入按钮，PLC 开始向变频器写入参数：电动机的斜坡上升时间定为 6s，在这段时间内，电动机加速至运行频率；斜坡下降时间定为 3s，在这段时间内，电动机减速至完全停止。当参数确定写入之后，按下起动按钮，变频器起动，6s 升到运行频率，本题运行频率有两档，分别是 25Hz 与 40Hz。停止时，按下停止按钮，变频器按照 PLC 为其设定的 3s 减速时间，从设定频率减少到 0，电动机停止。电动机可以正反转切换，但变频（25Hz 与 40Hz）可在运行中切换。

7. 通信电缆连接

如图 5-27 所示为整体连接，PLC 与变频器之间的连接还需特别明确一下，要求用一根带 D 型 9 针阳性插头的通信电缆接在 PLC（S7-200 PLC CPU226）的 0 号通信口，9 针并没有都用上，只接其中的 3 针，它们是 1（地）、3（B）、8（A），电缆的另一端是无插头的，以便接到变频器的 2、29、30 端子上，因这边是内置式的 RS-485 接口，参见图 5-26，在外面能看到的只是端子。两端的对应关系是 2↔1、29↔3、30↔8。连接方式示意图如图 5-28 所示。如 PLC 与变频器是点到点的连接，那么变频器这边还要接上偏置电阻，连接方式如图 5-29 所示。

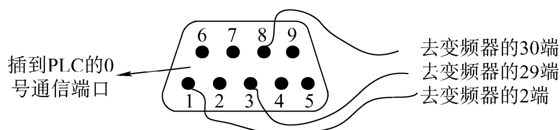


图 5-28　通信电缆的连接方式

8. 通信程序设计

首先对 USS 协议初始化，然后选择通信方式、地址、波特率等，根据从站地址判断控制对象。然后用写指令写入变频器的参数值。当确认参数全部写入变频

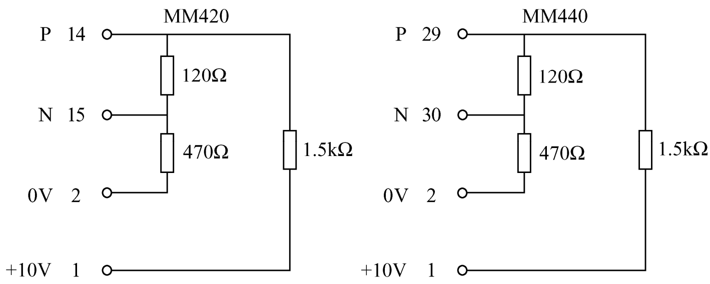


图 5-29 终端与偏置接线示意

器之后，起动变频器。变频器上升时间为 6s。变频器停止时，速度下降时间为 3s。变频器的 P1120 参数中的数据就是上升时间，单位为 s，所以把 6 写到 P1120 中即可；同样，P1121 是负责下降时间的，所以把 3 写到 P1121 中即可。用按钮进行运行频率（25Hz 与 40Hz）的切换，用按钮进行电动机正反转的切换。电动机减速停止的控制信号要求是连续的，也就是在整个减速过程中此信号应保持“ON”。快速停止可理解为“急停”，这个信号是脉冲信号即可。消除故障钮就是变频器故障复位钮，一般变频器出现内部故障后，先找出问题，排除故障，然后还需有一个外部信号，变频器接到此信号后，才能恢复工作。

（1）明确控制要求后，要进行 PLC 的 I/O 分配，PLC 的控制接线图如图 5-30 所示。

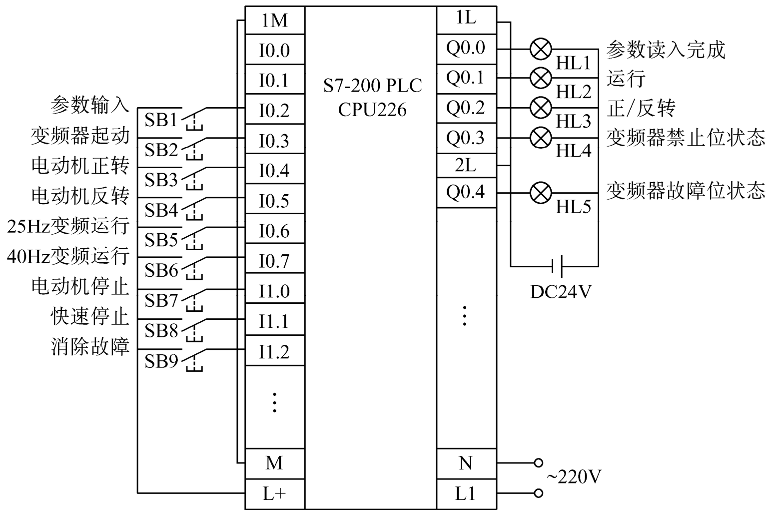


图 5-30 PLC 的控制接线图

（2）PLC 与变频器通信的控制程序梯形图及详解如图 5-31 所示。

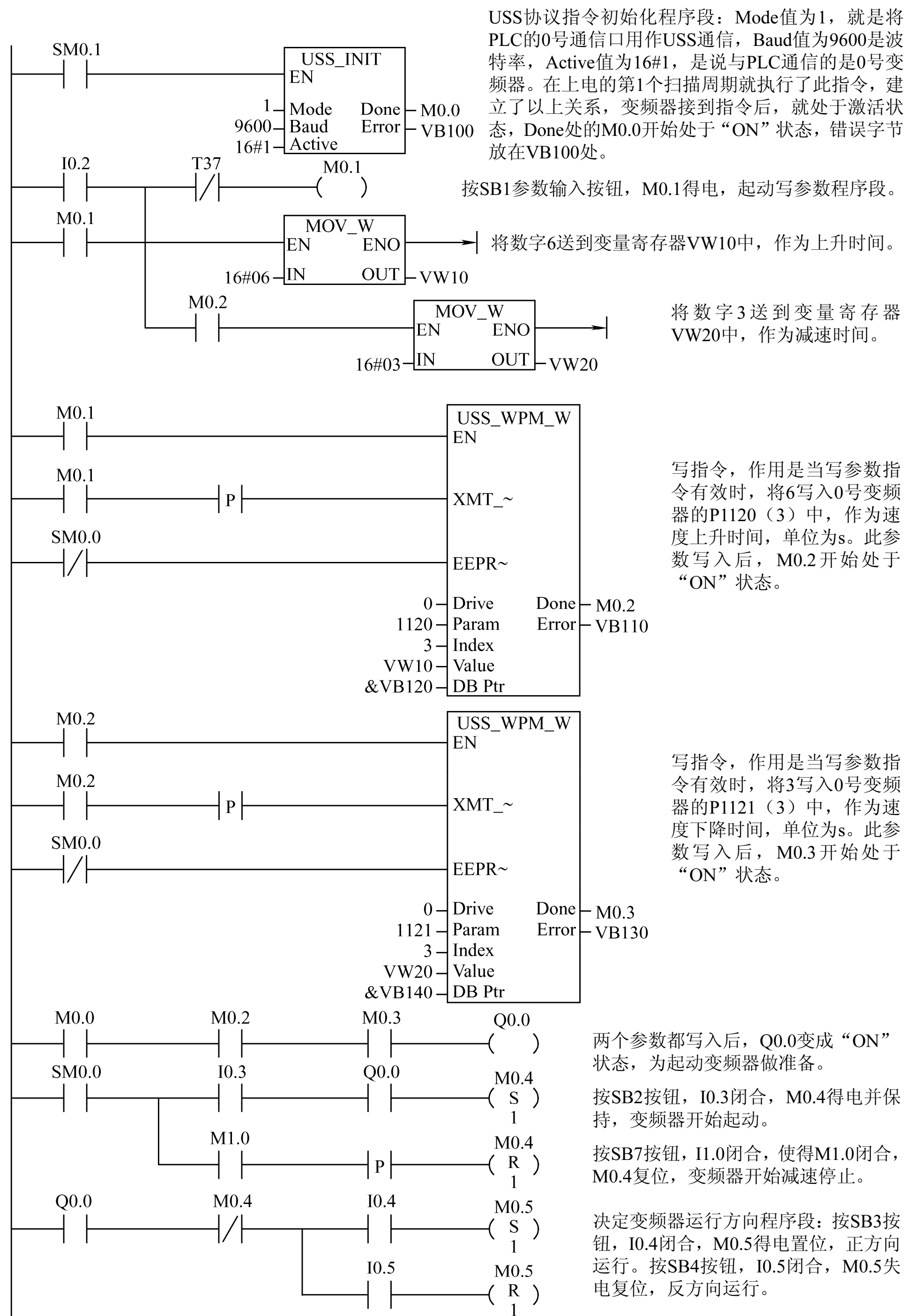


图 5-31　USS 通信指令控制程序梯形图

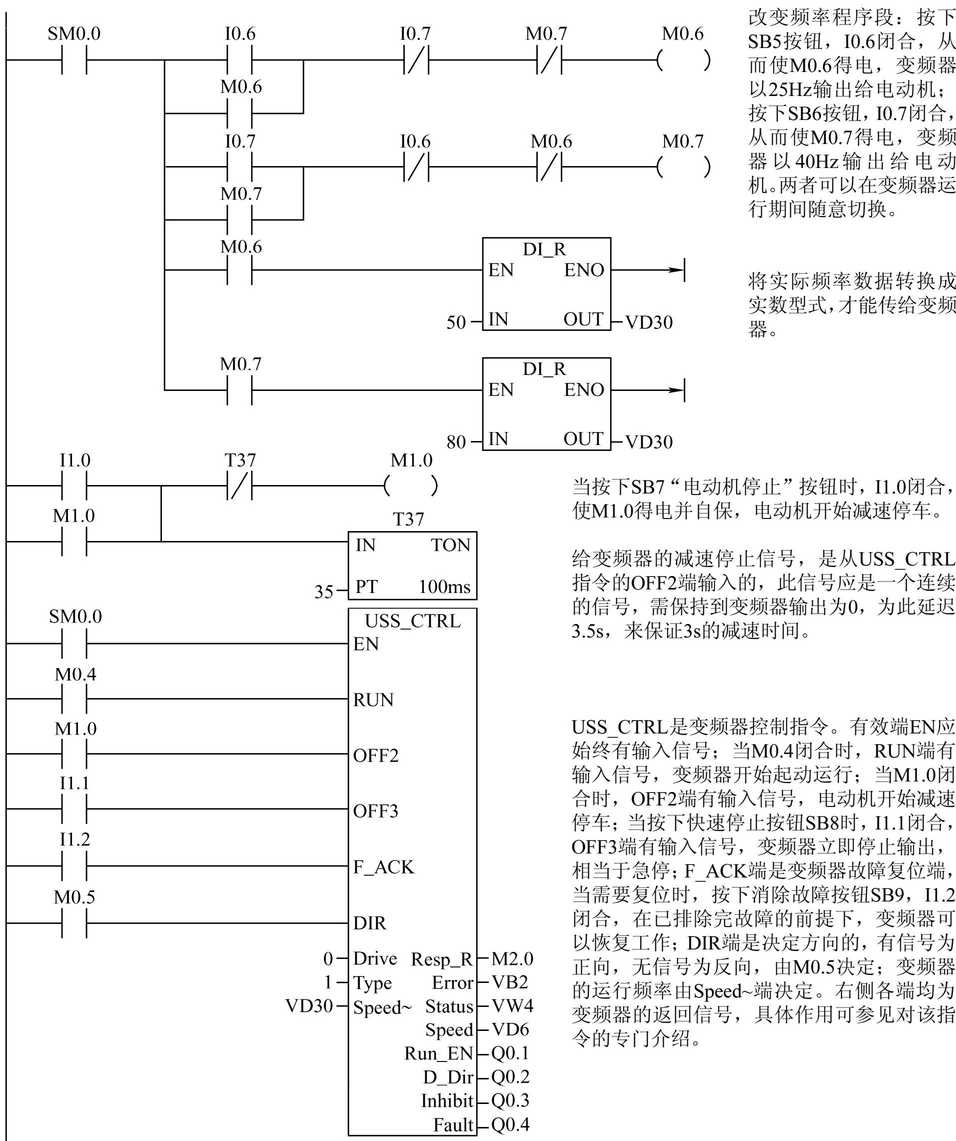


图 5-31 USS 通信指令控制程序梯形图（续）

例九、PLC 改造 B2012A 型龙门刨床控制系统

1. 改造意义

本题的改造目标为利用可编程序控制器及变频器实现对龙门刨床的自动控制和平滑调速，消除换向冲击，提高工作效率，减少噪声，取缔原控制系统，从而

达到经济快捷地运行龙门刨床的目的。使龙门刨床复杂的电气控制系统变得简单，清晰明了，使龙门刨床处于最佳的工作状态。龙门刨床如控制和使用得当，不仅能提高效率，节约成本，还可大大延长使用寿命。龙门刨床主要分为机械和电气控制两大组成部分，机械部分相对比较稳定，使龙门刨床运行在最优状态主要取决于电气控制系统控制方式。在传统龙门刨床中，其机械部分刚性好、精度较高，一般其基本性能可达到现代同类机械的水平，但控制和驱动部分则显出不同程度的老化，因此对老式龙门刨床的改造有很大的实际意义。

2. 工艺要求

本课题的工艺要求为

（1）取消电机扩大机、发电机，以减少噪音，克服诸多控制缺陷。

（2）工作台能实现自动循环工作和点动，可实时精确调节工作台速度，平稳换向，并有自动和点动工作时的极限保护。

（3）垂直刀架可方便地在水平和垂直两个方向快速移动和进刀，并能进行快速移动和自动进给的切换。

（4）左右侧刀架可在上、下方向快速移动和进刀，能进行快移/自动切换。并有左右侧刀架限位开关，防止其向上移动时与横梁碰撞。

（5）横梁可方便地上下移动和夹紧放松，夹紧程度可调；横梁下降时有回升延时，延时时间可调。

（6）润滑泵有连续/自动切换开关，系统一得电，油泵即上油，至一定压力时，油压继电器触点闭合，为工作台工作做准备。

（7）有保护环节控制，保证工作台停在后退末了，以免切削过程中发生故障而突然停车造成刀具损坏和影响加工工件表面的光洁度。

（8）各回路均有自动空气断路器作短路保护和过载保护。

3. 工作原理

B2012A 型龙门刨床接触器-继电器控制电路原理图如图 5-32 所示。

（1）主拖动机组的起动和停止　按主拖动机组的起动按钮 SB2，接触器 KM1 和 KMY 吸上。时间继电器 KT2 线圈得电，经延时后它的触点动作。接触器 KM1 的常开触点（703-705）闭合，实现自保。同时 KMY 的常闭触点（702-706）断开 KM2 与 KMΔ线圈通路，接触器 KMY 在主拖动机组定子侧的三个常开触点闭合，使主拖动机组接成 Y 起动。随着主拖动机组的起动，发电机 G 和励磁机 GE 也被拖动运转。当 GE 输出电压达到正常数值时，直流时间继电器 KT1 吸上，它的断电延时闭合的常闭触点（705-717）打开，断电延时打开的常开触点（723-725）闭合。由于时间继电器 KT2 延时时间尚未结束，它的触点（705-717）尚未断开，所以尽管 KT1 的触点（705-717）已打开，接触器 KMY 仍维持吸上状态，主拖动机组仍按星形联结运行。当时间继电器 KT2 的延时结束时，它的通电延时断开的常

闭触点（705-717）断开，接触器 KMY 断电释放，同时 KT2 的通电延时闭合的常开触点（705-723）和已经闭合的 KT1 的触点（723-725）共同使接触器 KM2 通电吸上。KM2 有两个常开触点闭合，一个实现自保（705-725），另一个（717-721）为接触器 KM△通电做好准备。KM2 的两个常闭触点断开，一个（717-719）使 KMY 线圈彻底失电，另一个 KM2 的常闭触点（31-51）使 KT1 线圈失电，在这里 KT1 是负责 Y 与△切换的间隔时间，经过整定延时后，KT1 的触点（705-717）闭合，触点（723-725）打开。这时 KM△线圈得电吸合，KM△在主拖动机组定子侧的三个常开触点闭合，使主拖动机组被连接成三角形运转，同时 KM△的常闭触点（702-704）断开 KT2 与 KMY 的线圈通路，到此主拖动机组起动完毕。

当按下主拖动机组的停止按钮 SB1 时，接触器 KM1、KM△和 KM2 均断电释放，主拖动机组的电源被切断机床停止工作。以上过程参见 5-32c 的 33～39 段。

（2）工作台的步进、步退　当主拖动机组起动完毕后，接触器 KM△的常开触点（101-103）闭合，按工作台步进按钮 SB8，继电器 KA3 通电吸合，KA3 的常开触点（1-3）闭合，断电延时继电器 KT3 吸合，KT3 的延时闭合的常闭触点（11-270）与（280-281）断开，断开了电机放大机的欠补偿回路和发电机的自消磁回路，同时 KT3 的延时断开的常开触点（1-201）与（2-204）闭合，使电机放大机控制绕组 WC3 中加入给定电压。

如放开按钮 SB8，它将自动复位，这时继电器 KA3 又断电释放，KA3 的常开触点（1-3）断开，KT3 断电，KT3 的延时断开的常开触点（1-201）与（2-204）延时断开，KT3 的延时闭合的常闭触点（11-270）与（280-281）延时闭合。电机放大机的欠补偿回路和发电机的自消磁回路被接通，工作台迅速制动下来，并防止了工作台的“爬行”。

工作台停车时，利用 KT3 的延时闭合的常闭触点，经过约 0.9s 的延时才接通电机放大机的欠补偿回路和发电机的自消磁回路，目的在于不使工作台停车制动过程过于强烈。

以上过程参见 5-32a 及 d 的 63～65 段。

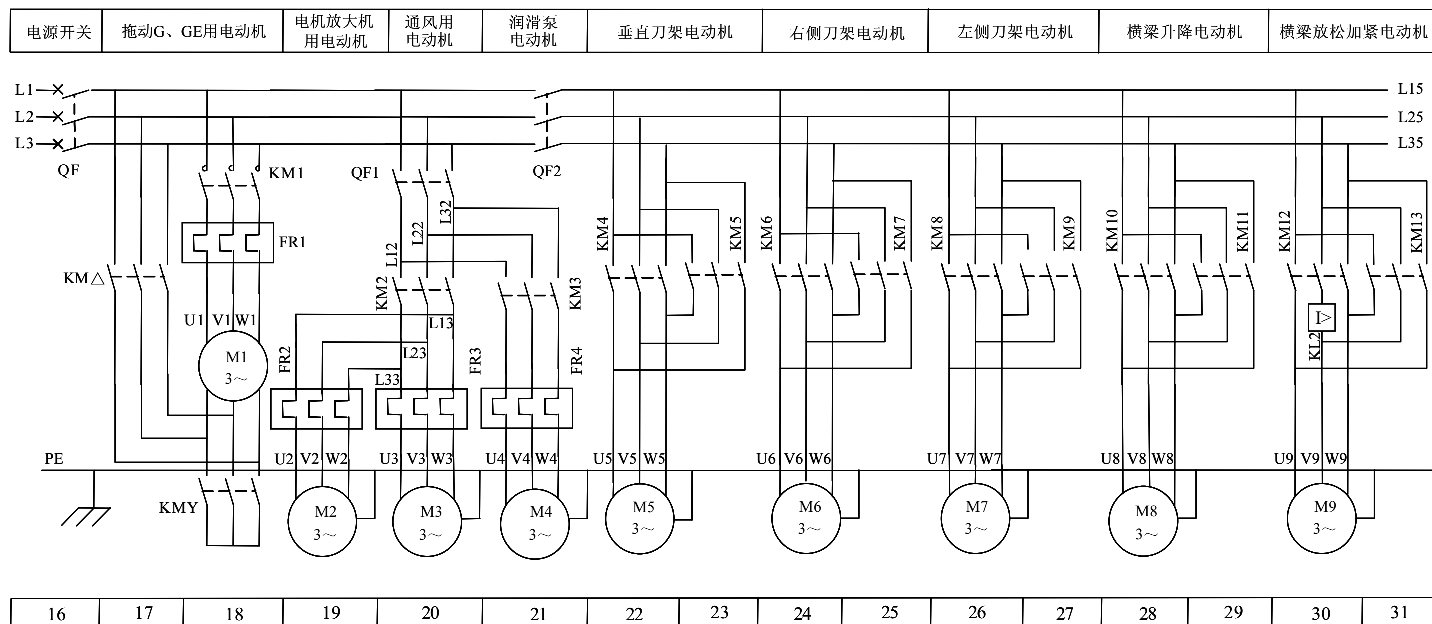
（3）工作台的自动循环工作　B2012A 龙门刨床在工作台侧面上装有四个撞块 A、B、C、D 在机床床身上装有四个行程开关 SQ3、SQ4、SQ6 及 SQ7 和两个终端限位开关 SQ5、SQ8。自动循环过程中的换速点就由 SQ3、SQ4、SQ6、SQ7 这四个行程开关的位置来决定。

假定工作台是停在返回行程末了的位置上，触点 SQ4-1（常闭）、SQ3-2（常闭）、SQ6-1（常开）、SQ7-2（常开）是闭合的，触点 SQ4-2（常开）、SQ3-1（常开）、SQ6-2（常闭）、SQ7-1（常闭）是断开的。按工作台前进按钮 SB9，继电器 KA4 得电吸上，KA4 的常开触点（107-129）闭合，实现自保，同时使继电器 KA6 得电吸上。KA4 的常开触点（111-113）闭合，继电器 KA3 得电吸上。KA4 的常

电机放大机（控制放大器）	发电机（功率放大器）	直流电动机（执行器、拖动刨台）	励磁发电机（直流电源）

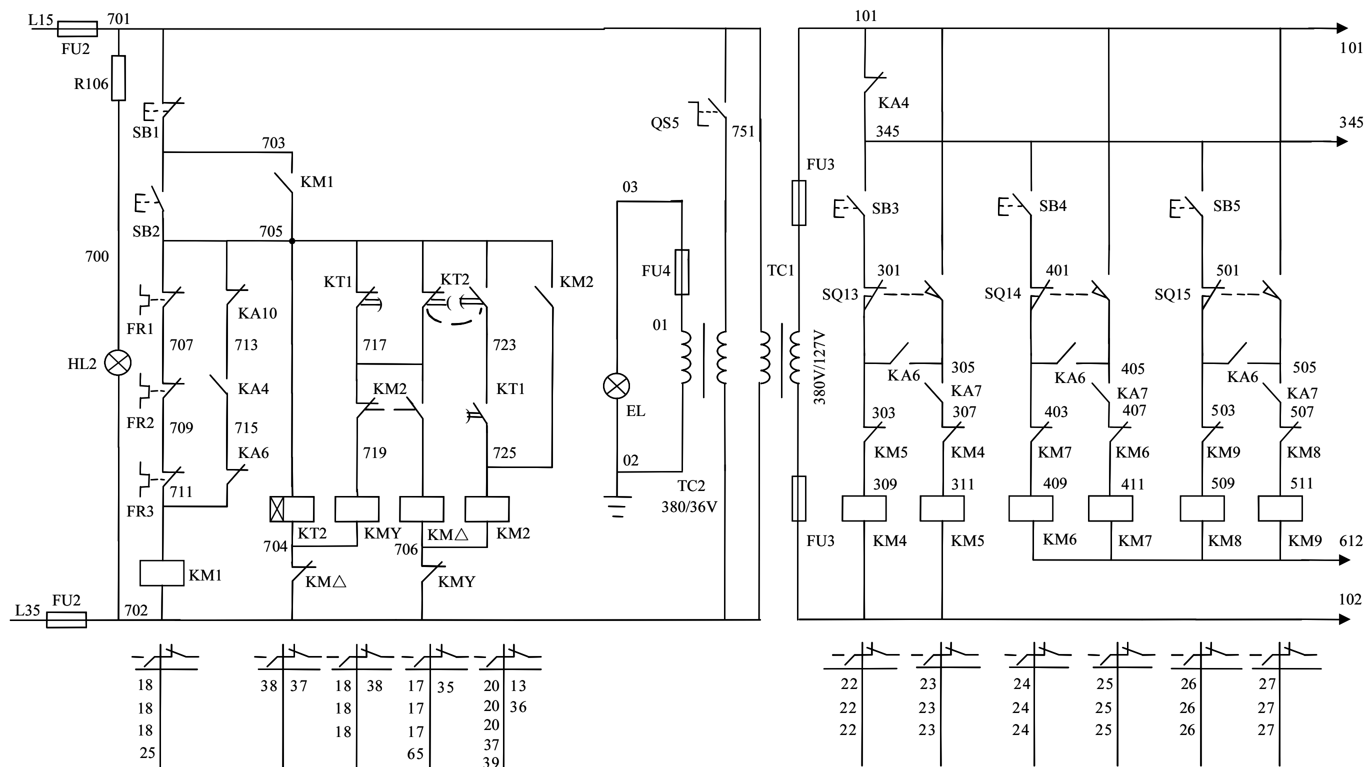
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

a）B2012A 型龙门刨床主拖动控制系统及抬刀电路原理图

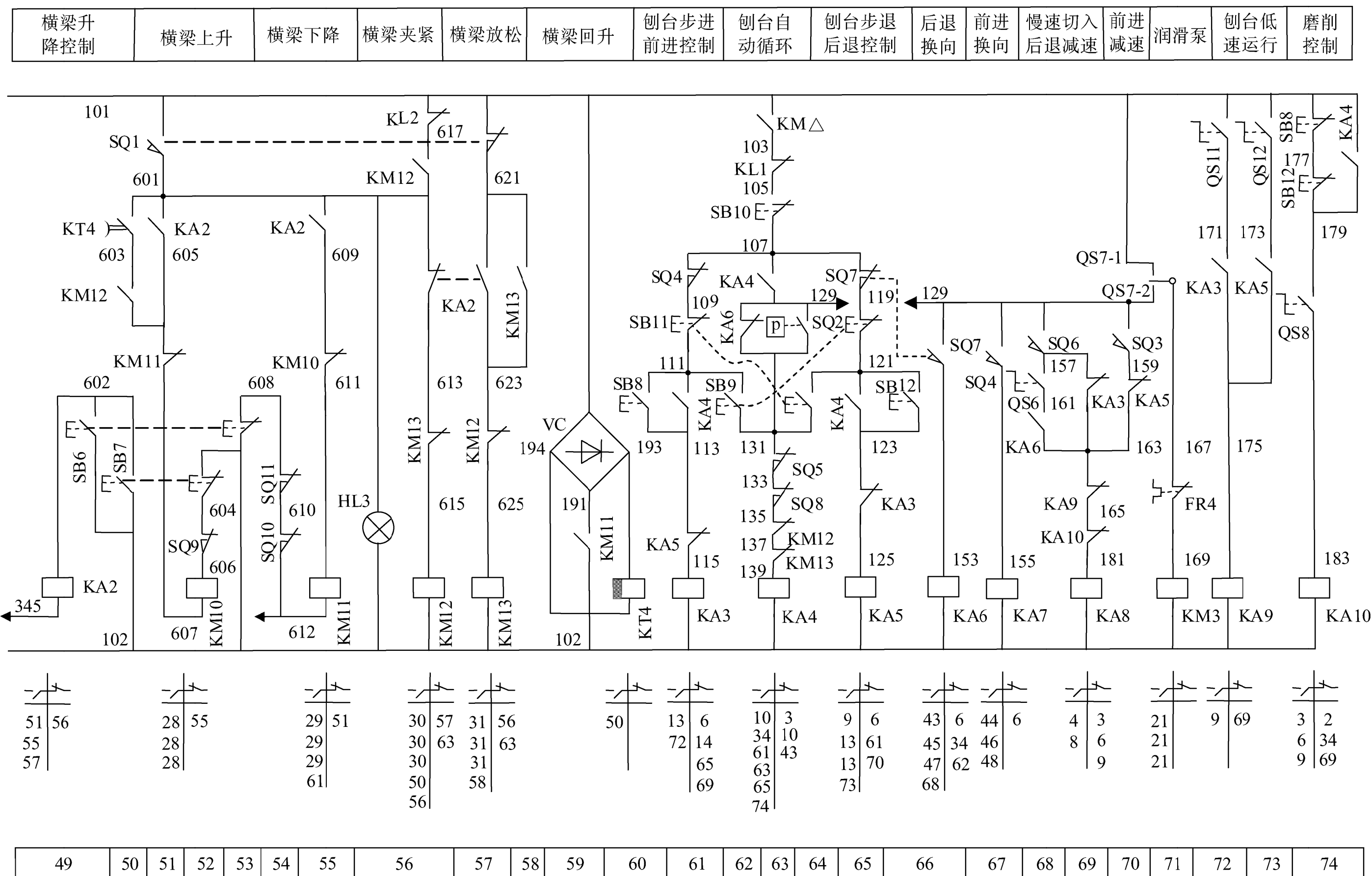


b）B2012A 型龙门刨床交流机组控制主电路原理图

图 5-32 B2012A 型龙门刨床接触器-继电器控制电路原理图



c）B2012A 型龙门刨床主拖动机组及刀架控制电路原理图



d）B2012A 型龙门刨床横梁及工作控制电路原理图

图 5-32　B2012A 型龙门刨床接触器-继电器控制电路原理图（续）

闭触点（200-240）断开，常开触点（200-220）闭合，使工作台调整回路断开，自动工作回路接通。

KA3 的常开触点（1-3）闭合，时间继电器 KT3 得电吸上，KT3 的延时闭合的常闭触点（11-270）及（280-281）断开，断开了电机放大机欠补偿回路和发电机的自消磁回路。KT3 的延时断开的常开触点（1-201）及（2-204）闭合，使调速电位器 RQ 及 RH 接通电源。由于转换开关 QS6 置于接通的位置上，在 KA6 的常开触点（161-163）闭合时，继电器 KA8 得电吸上，相当于工作台运行在慢速段。

继电器 KA3 及 KA8 的常开触点（220-225）、（225-237）闭合，KA8 的常闭触点（223-225）断开，电机放大机控制绕组 WC3 中便加入给定电压，电机放大机在强迫励磁作用下输出电压迅速升高达到稳定慢速时的数值，工作台因而也迅速起动并达到稳定的慢速。

工作台继续前进，撞块 D 使行程开关 SQ7 复位，触点 SQ7-1 闭合，为工作台反向做好准备。触点 SQ7-2 断开，继电器 KA6 断电释放，KA6 的常开触点（161-163）断开，继电器 KA8 断电释放。KA6 和 KA8 的常闭触点（230-250）和（250-281）恢复闭合状态。KA8 的常开触点（225-237）断开，常闭触点（223-225）闭合，断开了工作台的慢速回路，当撞块 C 碰撞行程开关 SQ6 时，SQ6 复位，触点 SQ6-1 断开，SQ6-2 闭合。工作台从此前进到高速段。在前进段快要结束时，撞块 A 碰行程开关 SQ3，触点 SQ3-1 闭合，继电器 KA8 又得电吸上，工作台又降到慢速运行。行程开关 SQ3-2 断开，将电阻 RbH 全部串入控制绕组 WC3 回路中，以限制减速、反向过程中主回路冲击电流不致过大，因而也减小了对传动机构部分的冲击。

当刀具离开工件，工作台工作（前进）行程结束时，撞块 B 碰行程开关 SQ4，触点 SQ4-1 断开，继电器 KA3 断电释放。触点 SQ4-2 闭合，继电器 KA7 得电吸上。KA3 的常开触点（200-225）断开控制绕组 WC3 正向励磁回路，KA3 的常闭触点（123-125）闭合，使继电器 KA5 得电吸上，此刻工作台后退（返回）已经开始。同时 KA3 的常闭触点（157-163）闭合，为工作台返回结束前的减速做好准备。

继电器 KA5 的常闭触点（159-163）断开，使继电器 KA8 断电释放，保证工作台以调速电位器 RH 的手柄位置所决定的高速返回，也即返回开始时没有低速段。KA5 的常开触点（220-226）闭合，接通了控制绕组 WC3 的反向励磁回路，工作台迅速制动并反向运行。同时 KA5 的常开触点（1-5）闭合，接触器 KMZ 得电吸上，KMZ 的常开触点（1-11）及（2-12）闭合，接通了抬刀电磁铁，刀架在工作台返回行程时，自动抬起。继电器 KA7 的常开触点（305-307）、（405-407）及（505-507）闭合，接通相应的接触器，使控制刀具的电动机反向旋转带着张紧环复位，为下一次的自动进刀做准备。

工作台以较高的速度返回，撞块 B 使行程开关 SQ4 复位时，触点 SQ4-1 闭合为工作台的正向运行做好准备。触点 SQ4-2 断开，继电器 KA7 断电释放，KA7 的常闭触点（210-230）闭合，切除串在控制绕组 WC3 回路的电阻 RbH 及 RbQ。KA7 的常开触点（305-307）、（405-407）及（505-507）断开，使相应的电机停止。

当撞块 A 使行程开关 SQ3 复位时，触点 SQ3-1 断开，SQ3-2 闭合。工作台返回行程将结束时，撞块 C 撞行程开关 SQ6，触点 SQ6-1 闭合，继电器 KA8 得电吸上，接通慢速回路，工作台改成慢速运行。在接近终点时，撞块 D 碰行程开关 SQ7，触点 SQ7-1 断开，继电器 KA5 断电释放，继电器 KA3 得电吸上。KA5 的常开触点（220-226）断开。KA3 的常开触点（220-225）闭合，同时，由于触点 SQ7-2 闭合，继电器 KA6 得电吸上，KA6 的常开触点（161-163）闭合，继电器 KA8 得电吸上，KA8 的常闭触点断开，常开触点闭合，控制绕组 WC3 中又加入正向给定电压，工作台迅速制动并立即正向起动，达到稳定的慢速，刀具在工作台慢速前进时切入工件，以后就重复上述运行过程，从而实现了工作台的往返自动循环工作。工作台自动往返的速度曲线图如图 5-33 所示。

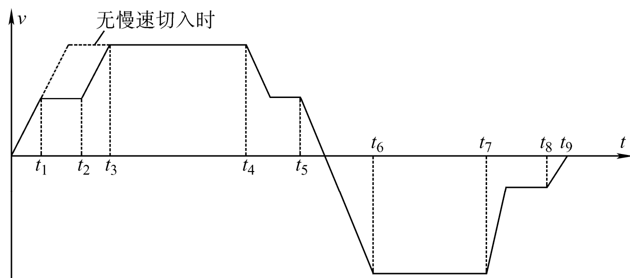


图 5-33　工作台带/不带慢速切入图

如果切削速度不太高，刀具能承受此时的冲击，或者是加工依次排列的短工件而无法利用“慢速切入”时，可以利用操纵台上的转换开关 QS6，将（157-161）断开，就可得到没有“慢速切入”的速度图，如图 5-33 的虚线部分所示。

当工作台速度低于 10m/min 时，触点 QS11（101-171）和 QS12（101-173）闭合，继电器 KA9 得电吸上，KA9 的常闭触点（163-165）断开，继电器 KA8 的回路切断，使“慢速切入”和换向前的减速环节均不起作用。

当机床用作磨削加工时，利用操纵台上的转换开关 QS8（179-183）接通，继电器 KA10 得电吸上，KA10 的常闭触点（165-181）断开继电器 KA8 的回路，使慢速环节不起作用。KA10 的常闭触点（201-203）断开，将电阻 RP11 串入，使给定电压减小，工作台降低到磨削时所要求的速度。同时在 RP8 上 KA10 的常开触点（4-B）和（290-294）闭合，加强了电桥稳定环节和电流正反馈环节的作用，使工作台在磨削加工时运行更加平稳，在负载变化时工作台的速度降落更小。

以上过程参见图 5-32a 及 c 的 43～48 段，d 的 63～77 段。

（4）刀架的控制　B2012A 型龙门刨床装有左侧刀架、右侧刀架和垂直刀架。这三个刀架分别采用三个交流电动机 M5、M6、M7 来拖动。

刀架的快速移动，自动进给及刀架运动的方向，由装在刀架进刀箱上的机械手柄来选择。刀架的进给机构采用张紧环，依靠张紧环旋转使张紧环复位，以便为第二次进刀做好准备。

当工作台按照工作行程前进，刀具离开工件，撞块 B 碰行程开关 SQ4 时触点 SQ4-2 闭合，继电器 KA7 通电吸上，KA7 的三个常开触点（305-307）、（405-407）、（505-507）闭合，使接触器 KM5、KM7、KM9 通电吸上，拖动三个刀架的电动机 M5、M6、M7 反转，带动张紧环复位，为进刀做好准备。由于触点 SQ4-1 断开，继电器 KA3 断电释放，KA3 的常闭触点（123-125）闭合，继电器 KA5 通电吸上，KA5 的常开触点（1-5）闭合，接触器 KMZ 通电吸上，KMZ 的常开触点（1-11）、（2-12）闭合，接通了抬刀电磁铁，刀架自行抬起。同时工作台制动并迅速返回。在工作台返回末了，撞块 D 碰行程开关 SQ7，触点 SQ7-1 断开，继电器 KA5 断电释放，KA5 的常闭触点（113-115）闭合，继电器 KA3 通电吸上，KA3 的常闭触点（5-7）断开了接触器 KMZ 回路，抬刀电磁铁断电释放，刀架放下，同时由于触点 SQ7-2 闭合，继电器 KA6 通电吸上，KA6 的常开触点（303-305）、（403-405）、（503-505）闭合，接触器 KM4、KM6、KM8 通电吸上，拖动三个刀架的电动机 M5、M6、M7 正转，并带动三个拨叉环旋转，完成三个刀架的进刀。

以上过程参见图 5-32a 的 12～15 段及 c 的 43～48 段，d 的 63～70 段。

（5）横梁升降的控制　横梁升降和放松，夹紧，分别用电动机 M8 和 M9 来拖动。按横梁上升按钮 SB6，继电器 KA2 得电吸合，它的常开触点（621-623）闭合，接触器 KM13 得电吸上，电动机 M9 反转，放松横梁，当横梁完全放松时，行程开关 SQ1 的触点 SQ1-2 断开，接触器 KM13 断电释放，电动机 M9 停止运转。同时由于触点 SQ1-1 闭合，接触器 KM10 得电吸上，电动机 M8 正转，横梁上升。当横梁上升到所需的位置放松按钮 SB6 时，继电器 KA2 断电释放，KA2 的常开触点（601-605）断开接触器 KM10，电动机 M8 停止，横梁停止上升，同时 KA2 的常闭触点（601-613）闭合，KM12 又得电吸上，电动机 M9 正转使横梁夹紧，同时行程开关 SQ1-1 断开，SQ1-2 恢复闭合状态。随着横梁的不断夹紧，电动机 M9 的电流也逐步增大，当横梁完全夹紧时，电流就增大到使电流继电器 KL2 动作的数值，KL2 吸上，当横梁完全夹紧时（101-617）断开，接触器 KM12 断电释放，电动机 M9 停止运转，横梁上升完毕。

当按横梁下降按钮 SB7 时，继电器 KA2 得电吸上，KA2 的常开触点（621-623）闭合，接触器 KM13 得电吸上，电动机 M9 反转，放松横梁，当横梁完全放松时，行程开关 SQ1 闭合，它的触点 SQ1-2 断开，接触器 KM13 断电释放，电动机 M9

停止运转。同时由于触点 SQ1-1 闭合，接触器 KM11 得电吸上，电动机 M8 反转。横梁下降，KM11 的常开触点（101-191）闭合，延时释放继电器 KT4 得电吸上，KT4 的延时断开的常开触点（603-605）闭合，为横梁下降后回升做好准备。

当横梁下降到需要的位置放开按钮 SB7 时，继电器 KA2 和接触器 KM11 断电释放，电动机 M8 停止运转，横梁不再下降。同时由于 KM11 的常开触点（101-191）断开，继电器 KT4 断电延时释放。又由于 KA2 的常闭触点（601-613）闭合，接触器 KM12 得电吸上，电动机 M9 正转使横梁夹紧。KM12 的常开触点（601-603）闭合，接触器 KM10 得电吸上，电动机 M8 正转，使横梁在夹紧的过程中同时回升。当继电器 KT4 的常开延时断开触点（603-605）断开时，横梁回升停止。

以上过程参见图 5-32d 的 49～62 段。

4. 程序设计

（1）龙门刨床电气原理图新旧代号对照及 PLC 的 I/O 分配见表 5-2。

（2）日本安川 616R3 型变频器对外接线端子分配及速度与端子的关系。

表 5-2　龙门刨床电气原理图新旧代号对照及 PLC 的 I/O 分配

I 分配				
序 号	点	代 号	原代号	注 释
1	I0.0	KL2	JL-J	横梁夹紧电动机电流继电器
2	I0.1	SB2	3A	垂直刀架快速移动
3	I0.2	SB3	4A	右侧刀架快速移动
4	I0.3	SB4	5A	左侧刀架快速移动
5	I0.4	SB5	6A	横梁上升
6	I0.5	SB6	7A	横梁下降
7	I0.6	SQ1	6HXC	横梁放松到位才能上升或下降
8	I0.7	SB7	8A	工作台步进
9	I1.0	SB8	9A	工作台前进
10	I1.1	SB9	10A	工作台停止
11	I1.2	SB10	11A	工作台后退
12	I1.3	SB11	12A	工作台步退
13	I1.4	QS6	6KK	慢速切入开关
14	I1.5	QS7	7KK	润滑泵接通开关
15	I1.6	QS8	8KK	磨削运行时接通开关
16	I1.7	QS11（12）	KK-Q（H）	低速运行时接通开关
17	I2.0	BP		变频器零速

（续）

I 分配				
序 号	点	代 号	原代号	注　　释
18	I2.1	SQ2	Je	润滑油油压开关
19	I2.2	SQ3	Q-JS	前进减速限位开关
20	I2.3	SQ4	Q-HX	前进换向限位开关
21	I2.4	SQ5	1HXC	工作台前进终端限位
22	I2.5	SQ6	H-JS	后退减速限位开关
23	I2.6	SQ7	H-HX	后退换向限位开关
24	I2.7	SQ8	2HXC	工作台后退终端限位
O 分配				
序 号	点	代 号	原代号	注　　释
1	Q0.0	YA1	1T	右侧刀架抬入电磁铁控制
2	Q0.1	YA2	2T	左侧刀架抬入电磁铁控制
3	Q0.2	YA3	3T	右侧垂直刀架抬入电磁铁控制
4	Q0.3	YA4	4T	左侧垂直刀架抬入电磁铁控制
5	Q0.4	KM4	Q-C	垂直刀架快速移动及正常进给
6	Q0.5	KM5	H-C	垂直刀架正常工作时反转复位
7	Q0.6	KM6	Q-Y	右侧刀架快速移动及正常进给
8	Q0.7	KM7	H-Y	右侧刀架正常工作时反转复位
9	Q1.0	KM8	Q-Z	左侧刀架快速移动及正常进给
10	Q1.1	KM9	H-Z	左侧刀架正常工作时反转复位
11	Q1.2	KM3	C-RB	润滑油泵
12	Q1.3	KM13	H-J	横梁放松
13	Q1.4	KM12	Q-J	横梁夹紧
14	Q1.5	KM10	Q-H	横梁上升
15	Q1.6	KM11	H-H	横梁下降
16	Q1.7	KM16		给变频器
17	Q2.0			正转
18	Q2.1			反转
19	Q2.2			速度端口 1
20	Q2.3			速度端口 2
21	Q2.4			速度端口 3

（3）PLC 的电气控制系统接线图如图 5-34 所示。

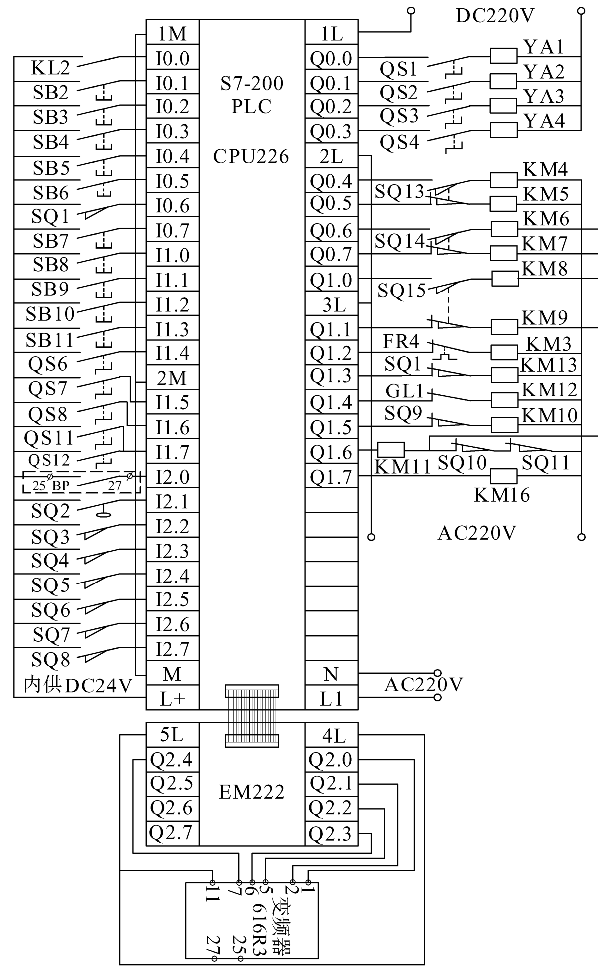


图 5-34　PLC 电气控制系统接线图

（4）程序编辑与注释说明。

本次改造不包括原控制系统电路中的指示灯、仪表等。除主拖动机组外其他电动机主电路接线与改造前相同。

变频器的对外接线图如图 5-35 所示，变频器的 R、S、T 为三相电源输入端；U、V、W 为三相变频输出端（至电动机）。1、2 号端子为电机的正转和反转；5、6、7 号端子为电机的三个段速；25、27 号端子为变频器的零速输出。均由 PLC 控制，以实现正转、反转、减速停止等运行及不同速度的输入。该刨床要求有 8 种速度：①正常切削速度大约为 60m/min；②正常切削返回速度大约为 80m/min；

③正常切削时的慢速段速度大约为 12m/min；④步进、步退速度大约为 3～8m/min；⑤低速切削速度为 10m/min；⑥磨削速度为 1m/min；⑦低速返回速度大约为 40m/min；⑧磨削返回速度（同③）。速度的调整通过改变变频器 5、6、7 号输入端的输入组合来实现。变频器内部决定速度的参数与接线端子的状态是一一对应的，在这里不再赘述。其速度值输出分配如表 5-3 所示。

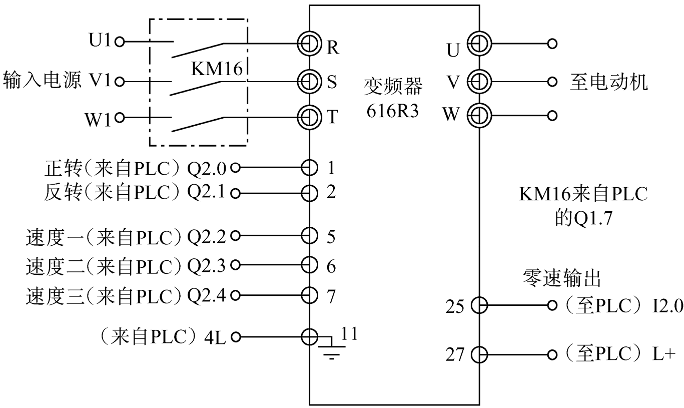


图 5-35　变频器的对外接线图

表 5-3　速度值输出分配

端子＼段速	①	②	③	④	⑤	⑥	⑦	⑧
5 端	1	0	0	1	1	0	1	同③
6 端	0	1	0	1	0	1	1	
7 端	0	0	1	0	1	1	1	

```
主程序                       // 程序注释
Network 1                    // 变频器起停控制程序段
LD      I0.7                 // 工作台步进
O       I1.0                 // 工作台前进
O       I1.2                 // 工作台后退
O       I1.3                 // 工作台步退
EU
LD      T38                  // 停车后需延迟时间才能断开变频器电源
NOT
LPS
A       Q1.7
=       Q1.7                 // 给变频器提供电源的接触器置位
```

```
LPP
ALD
O         Q1.7
=         Q1.7          // 给变频器提供电源的接触器复位
Network 2               // 起动后的延迟与停止后的延迟程序段
LD        Q1.7
LPS
AN        M0.0
TON       T37, 20       // 变频器起动后延迟 2s 工作台才可工作
LRD
LD        T37
O         M0.0
ALD
AN        I1.1
=         M0.0          // 2s 后此继电器闭合工作台就可工作了
LPP
A         I2.0          // 变频器已停止运转，零速信号已到
AN        M0.0          // 工作台总控继电器已失电
AN        Q2.0          // 变频器没有正向运行信号输出
AN        Q2.1          // 变频器没有反向运行信号输出
TON       T38, 30       // 经 3s 后断开变频器供电电源
Network 3               // 工作台自动循环主控程序段
LD        M0.0          // 变频器起动完毕
LD        I1.0
AN        I1.2
AN        I2.3          // 工作台前进起动段
LD        I1.2
AN        I1.0
AN        I2.6          // 工作台后退起动段
OLD
LD        I2.1          // 润滑油压力开关必须闭合
ON        I2.6          // 即使没有润滑油也要继续工作到后退段的末端
A         M0.4          // 总控的自锁触点
OLD
ALD
```

```
AN        I2.4          // 前进终端限位
AN        I2.7          // 后退终端限位
AN        Q1.3          // 横梁夹紧时工作台不能自动循环
AN        Q1.4          // 横梁放松时工作台不能自动循环
=         M0.4          // 工作台自动循环总控继电器
Network 4               // 变频器运行速度控制程序段
LD        M0.0          // 变频器起动完毕
LPS
LD        I0.7          // 工作台步进工作按钮
O         M0.4          // 工作台自动循环工作
ALD
AN        I1.2          // 前进时不能按后退按钮
AN        I2.3          // 前进换向限位开关
AN        Q2.1          // 前进与后退的互锁
=         Q2.0          // 变频器前进方向信号
LRD
LD        I1.3          // 工作台步退工作按钮
O         M0.4          // 工作台自动循环工作
ALD
AN        I1.0          // 后退时不能按前进按钮
AN        I2.6          // 后退换向限位开关
AN        Q2.0          // 前进与后退的互锁
=         Q2.1          // 变频器后退方向信号
LRD
LD        M0.4          // 工作台以自动循环方式工作
LD        Q2.0          // 前进方向
A         I2.0          // 变频器零速状态
O         Q2.2          // 段速信号自锁
AN        I1.4          // 无慢速切入工作状态
AN        I2.2          // 前进减速限位开关，撞到此开关即减速
LD        Q2.0          // 前进方向有慢速切入工作状态
AN        I2.2
AN        I2.6          // 前进刚开始的慢速段此限位开关断开此通路
OLD
LD        Q2.0          // 前进或后退的低速工作方式
```

```
O       Q2.1
A       I1.7          // 低速方式
OLD
ALD
LDN     M0.4          // 以下为步进、步退工作段，自动循环控制断开
LD      Q2.0
A       I0.7          // 步进按钮
LD      Q2.1
A       I1.3          // 步退按钮
OLD
ALD
OLD
ALD
=       Q2.2          // 给变频器段速信号的第一端口
LRD
LD      M0.4          // 以下工作过程受自动循环控制
LD      Q2.0          // 前进方向
A       I1.6          // 磨削方式
AN      I0.7          // 不能按步进按钮
AN      I1.3          // 不能按步退按钮
LD      Q2.1          // 后退方向
A       I2.0          // 由前进变成后退应等待变频器零速信号
O       Q2.3          // 段速信号自锁
AN      I2.5          // 后退减速限位开关，撞到此开关即减速
OLD
LD      Q2.1          // 后退方向
A       I1.7          // 低速方式
OLD
ALD
LDN     M0.4          // 以下工作过程不受自动循环控制
LD      I0.7          // 步进按钮
A       Q2.0
LD      I1.3          // 步退按钮
A       Q2.1
OLD
```

```
ALD
OLD
ALD
=         Q2.3        // 给变频器段速信号的第二端口
LPP
A         M0.4        // 以下工作过程受自动循环控制
LD        Q2.0        // 前进方向
A         I2.0        // 由后退变成前进应等待变频器零速信号
O         Q2.4        // 变频器第三段速口信号自锁
A         I1.4        // 有慢速切入工作方式
A         I2.6        // 从后退末端换向点开始闭合到前进方向后断开
LD        I2.5        // 后退到减速限位开关时此点闭合直到转为前进后复位
AN        Q2.0        // 前进方向
LD        I2.2        // 前进到减速限位开关时此点闭合直到转为后退后复位
AN        Q2.1        // 后退方向
OLD
OLD
AN        I1.6        // 不是低速工作模式
AN        I1.7        // 不是磨削工作模式
LD        Q2.0
O         Q2.1        // 不管前进或后退以下程序都起作用
LDN       I0.7
AN        I1.3        // 不能有步进、步退工作方式
A         I1.6        // 磨削工作方式开关已接通
O         I1.7        // 低速工作方式开关已接通
ALD
OLD
ALD
=         Q2.4        // 给变频器段速信号的第三端口
Network 5             // 刀架上抬刀电磁铁及刀架自身进给控制程序段
LD        SM0.0
LPS
AN        Q2.0
A         Q2.1        // 只要后退方向信号到来刀架，就开始抬起直到换向
S         Q0.0, 4     // 给四个刀架上电磁铁的抬刀信号
```

```
LRD
AN      Q2.1
A       Q2.0
R       Q0.0, 4        // 运行方向换为前进后，已抬起的刀架又垂下复位
LRD
LDN     M0.4           // 以下动作在非自动循环状态下进行
A       I0.1           // 垂直刀架快速移动按钮
LD      M0.4           // 以下动作在自动循环状态下进行
A       I2.6           // 工作台后退到后退换向限位开关时，此点开始闭合
OLD
ALD
AN      Q0.5           // 垂直刀架进给与刀架后退复位的互锁
=       Q0.4           // 垂直刀架带着刀具进给到规定的行程开始进行刨削
LRD
LDN     M0.4           // 以下动作在非自动循环状态下进行
A       I0.2           // 右侧刀架快速移动按钮
LD      M0.4           // 以下动作在自动循环状态下进行
A       I2.6           // 工作台后退到后退换向限位开关时，此点开始闭合
OLD
ALD
AN      Q0.7           // 右侧刀架进给与刀架后退复位的互锁
=       Q0.6           // 右侧刀架带着刀具进给到规定的行程开始进行刨削
LRD
LDN     M0.4           // 以下动作在非自动循环状态下进行
A       I0.3           // 左侧刀架快速移动按钮
LD      M0.4           // 以下动作在自动循环状态下进行
A       I2.6           // 工作台后退到后退换向限位开关时，此点开始闭合
OLD
ALD
AN      Q1.1           // 左侧刀架进给与刀架后退复位的互锁
=       Q1.0           // 左侧刀架带着刀具进给到规定的行程开始进行刨削
LPP
A       M0.4           // 以下动作在自动循环状态下进行
A       I2.3           // 工作台前进到前进换向限位开关时，此点开始闭合
LPS
```

```
AN      Q0.4        // 垂直刀架进给与刀架后退复位的互锁
=       Q0.5        // 垂直刀架后退复位为下一个进刀动作做准备
LRD
AN      Q0.6        // 右侧刀架进给与刀架后退复位的互锁
=       Q0.7        // 右侧刀架后退复位为下一个进刀动作做准备
LPP
AN      Q1.0        // 左侧刀架进给与刀架后退复位的互锁
=       Q1.1        // 左侧刀架后退复位为下一个进刀动作做准备
Network 6           // 横梁升降、放松夹紧控制程序段
LD      SM0.0
LPS
LD      I0.4        // 横梁上升按钮
O       I0.5        // 横梁下降按钮
ALD
=       M0.5        // 横梁升降中间控制
LRD
LD      M0.5
O       Q1.3        // 横梁放松自锁控制
ALD
AN      I0.6        // 横梁放松到位此点断开
AN      Q1.4        // 横梁夹紧与放松互锁
=       Q1.3        // 横梁放松
LRD
LDN     I0.0        // 横梁夹紧电动机电流继电器，夹紧到位时动作断开
A       Q1.4        // 横梁夹紧动作时此点自锁
O       I0.6        // 横梁放松到位后才能有夹紧动作
ALD
LPS
AN      M0.5        // 横梁夹紧过程中不能按横梁上升、下降按钮
AN      Q1.3        // 横梁夹紧过程中不能有横梁放松动作
=       Q1.4        // 横梁夹紧
LRD
LD      Q1.4        // 在横梁夹紧过程中可以完成下面的动作
A       T39         // 横梁下降后为防丝杠变形还有短暂的上升时间段
O       M0.5        // 横梁上升控制按钮
```

```
ALD
AN    I0.5       // 横梁上升期间不能按横梁下降按钮
AN    Q1.6       // 横梁上升与横梁下降的互锁
=     Q1.5       // 横梁上升
LPP
A     M0.5       // 横梁下降动作
AN    I0.4       // 横梁下降期间不能按横梁上升按钮
AN    Q1.5       // 横梁上升与横梁下降的互锁
=     Q1.6       // 横梁下降
LPP
A     Q1.6       // 在横梁下降时接通 T39 断电延迟型时间继电器
TOF   T39, 30    // 横梁下降后的上升动作段定时器
Network 7        // 油泵控制程序段
LD    M0.4       // 在自动循环工作方式下
A     I1.5       // 润滑油泵工作方式定为自动循环
ON    I1.5       // 润滑油泵工作方式定为直接自行工作
=     Q1.2       // 润滑油泵接触器线圈
```

例十、波浪式喷泉的PLC控制程序

1. 控制要求

在一些休闲、娱乐、旅游景点，经常会修建喷泉供人们观赏。这些喷泉按一定的规律改变喷水式样，有的像花朵，有的可形成水幕放电影，有的可随着音符跳跃，形式多样。本题所控制的喷泉是波浪式的，可用在湖面上，从远处看，给人的感觉像是湖面上掀起了波浪，示意图如图 5-36 所示。按下起动按钮后，喷泉开始运作，共有 3 个波峰，1 个波峰为 1 组，1 组有 5 个喷头，这样总共有 15 个喷头，某一时刻只有 1 组在工作，按 1、2、3 顺序排队，形成移动的波浪。而每组在运作时也要按一定的规律有先有后。在本组内的 5 个喷头是这样工作的，每隔 3s 开启 1 个，到第 4 个开启时同时关闭第 1 个，到第 5 个开启时同时关闭第 2 个，3s 后下一组开始工作，前面一组全部关闭。如此，3 个组按顺序循环工作，直到按下停止按钮，全部喷头都停止工作。从按下起动按钮，到一个工作周期结

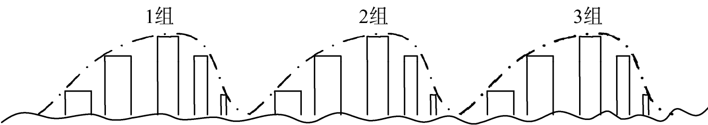


图 5-36　湖面上喷头组别位置示意图

束，各喷头工作状态时序图如图 5-37 所示。

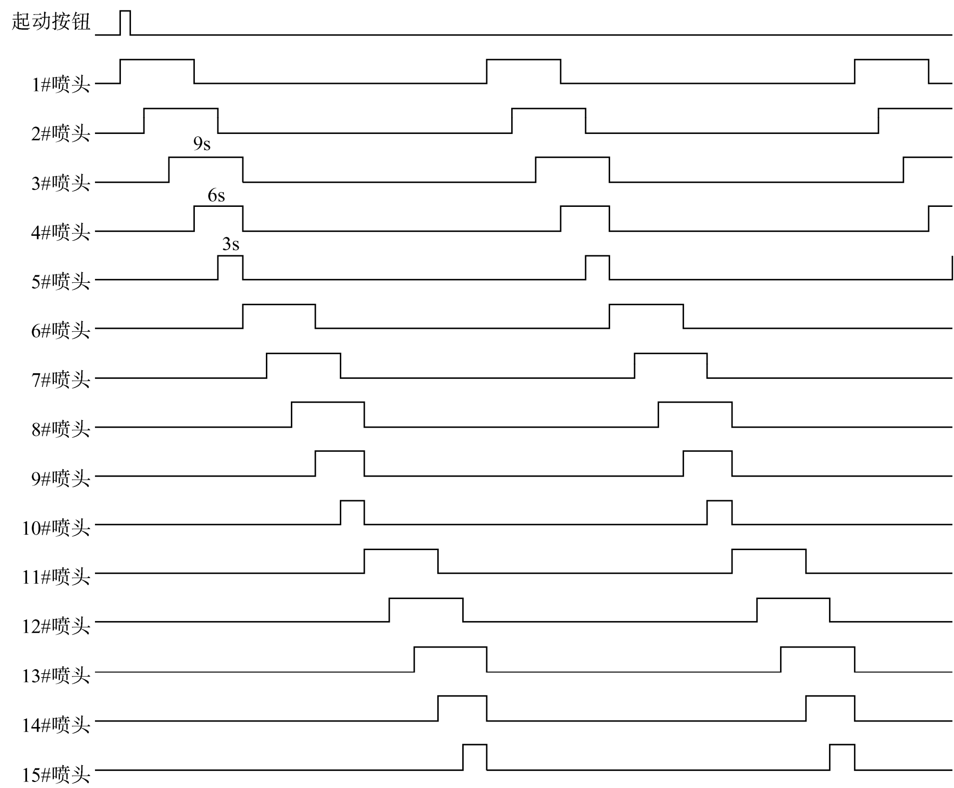


图 5-37　各喷头工作状态时序图

2. 程序设计

（1）根据控制要求，首先决定 PLC 的 I/O 分配，如图 5-38 所示。

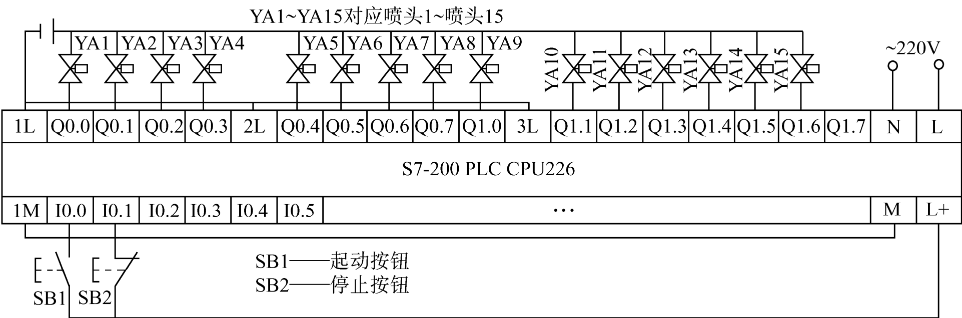


图 5-38　PLC 的电气控制系统接线图

（2）本题 15 个喷头分成 3 组，每组 5 个按顺序起停，3 个组的工作过程都是

一样的，参见图 5-37。按起动按钮后，喷头就会按要求动作，整个过程是自动循环的，只有按下停止按钮，才会全部停止。程序设计准备以移位寄存器指令 SHRB 为主，程序中还多次出现比较指令用定时器的当前值与整数比较，这也是以往未被重用的一个功能。

（3）控制程序梯形图及注释如图 5-39 所示。

I0.0　I0.1　M0.0
M0.0　M0.4　T37　IN　TON
600　PT　100ms

按下起动按钮SB1，I0.0闭合，M0.0得电并自保，喷泉开始工作，按下停止按钮SB2，I0.1断开，喷泉停止工作。起动后，T37开始全程计时

T37 ==I 30　P　M0.1
计时到30s，定Q0.1的位状态
T37 ==I 60
计时到60s，定Q0.2的位状态
T37 ==I 180　P　M0.2
计时到180s，定Q0.6的位状态
T37 ==I 210
计时到210s，定Q0.7的位状态
T37 ==I 330　P　M0.3
计时到330s，定Q1.3的位状态
T37 ==I 360
计时到360s，定Q1.4的位状态

由M0.1负责第1组、M0.2负责第2组、M0.3负责第3组延迟移位时的位状态，每隔3s，喷泉的工作状态都会变化，变化是由位的状态决定的，而决定位状态的是M0.1、M0.2、M0.3

I0.0　P　MOV_W　EN　ENO
M0.5　16#0100　IN　OUT　QW0

按下起动按钮后，首先将1送到QW0中，也就是先将Q0.0定为1，这时第1个喷头将开始工作。M0.5是负责循环的

T37 ==I 30　P　SHRB　EN　ENO
T37 ==I 60　M0.1　DATA
Q0.0　S_BIT
5　N
T37 ==I 90
计时到90s，定Q0.3的位状态
T37 ==I 120
计时到120s，定Q0.4的位状态

每隔3s移位1次，每组移位长度为5，第1组为Q0.0~Q0.4。移位时，新进来的数据是1还是0，第1组由M0.1当时的状态决定，如果M0.1是1，位的状态就是1；M0.1是0，位的状态就是0

图 5-39　控制程序梯形图及注释

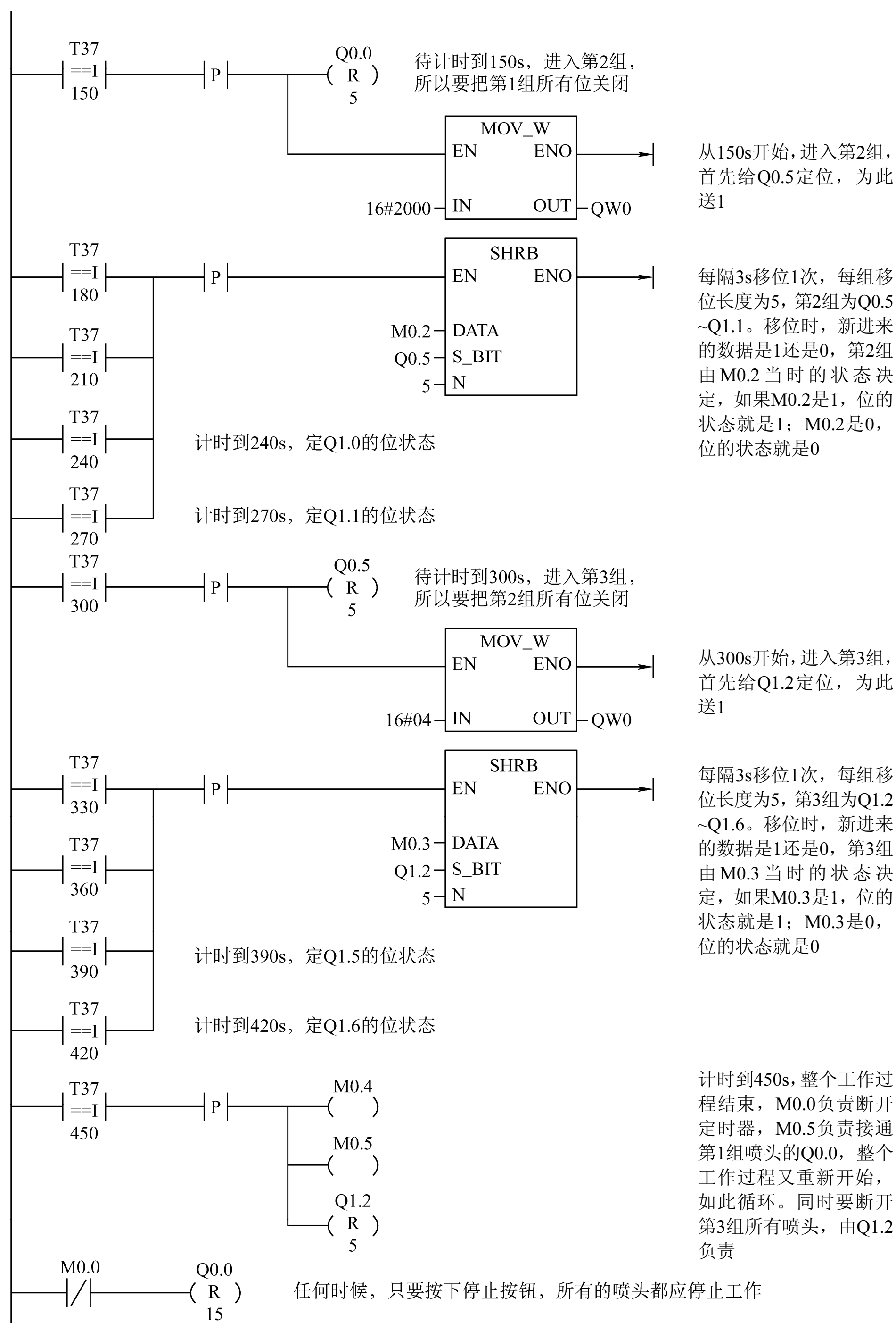


图 5-39　控制程序梯形图及注释（续）

例十一、变频器调速系统远程监控的实现

1. 什么是远程监控

其中“监”也就是监视，监视的内容很多，这里是指对设备运行状况的监视，这个监视不是摄像头视频监视，而是在计算机上制作一个动态画面，这个画面完全模拟现场设备的动作，比如设备上的机械手抓起了一个被加工件，画面上也应同步出现这个动作；而“控”也就是控制，控制的方式很多，这里是指通过通信网络，在计算机上进行操作来控制设备，在计算机画面上画一个按钮，可以让它动起来并控制现场设备的起/停；“远程”是指监视与控制生产设备的监控者并没有在现场，是有距离的，这个距离就不好说了，也许仅有几米，也许在异国他乡。当今 PLC 的功能已经很强了，但是没有人机联系的界面，动态画面看不见。由此而产生了很多可以在计算机上制作画面的软件，实现远程监控。这个画面不是静止不动的，经过一番设置，画面上的图素可以动起来，从而反映出现场设备的某个具体动作。这个软件与现场的智能设备是可通信的、相互支持的，比如 PLC，它们相互间可传递数据。

2. Kingview 组态王软件简介

组态是用应用软件中提供的工具、方法来完成工程中某一具体任务的过程。与硬件生产相对照，组态与组装类似，但软件中的组态要比硬件的组装有更大的发挥空间，一般要比硬件中的“部件” 更多，而且每个“部件”都很灵活，因软部件都有内部属性，通过改变属性可以改变其规格（如大小、性状、颜色等）。

组态软件是有专业性的，一种组态软件只能适合某种领域的应用，人机界面生成软件就叫工控组态软件，组态结果是用在实时监控的。这样的软件填补了 PLC 的“美中不足”。

组态不需要编写程序就能完成特定的应用，但是为了提供一些灵活性，组态软件也提供了编程手段，一般都是内置编译系统。

Kingview 组态王是北京亚控科技发展有限公司开发的一种组态软件，它可以运行在当今的个人电脑的操作系统下，版本不断翻新。可与可编程控制器（PLC）、智能模块、智能仪表、板卡、变频器等多种外部设备进行通讯。而其软件系统与用户最终使用的现场设备无关，比如本课题看似是用实时曲线监控变频器的调速过程，实际上组态软件跟现场的变频器一点联系都没有，与它联系的是 PLC。对于不同的硬件设施，用户只需要按照安装向导的提示完成 I/O 设备的配置工作，比如本课题用的是西门子 S7-200 PLC，我们可以按照提示一步一步完成配置，只要配置准确，组态软件就可与 PLC 之间进行通信了。

组态王主要有以下几种功能：使用清晰准确的画面描述工业控制现场，使用图形化的控制按钮实现单任务和多任务，设计复杂的动画，显示现场的操作状态

和数据，显示生产过程的文字信息和图形信息，为任何现场画面指定键盘命令，监控和记录所有报警信息，显示实时趋势曲线和历史趋势曲线，等等。利用“组态王”对 PLC 进行动画组态、硬件组态和控制组态，将二者结合起来实现整个生产过程的综合监控。

3. 控制要求

- 变频器选用西门子 Micro Master 420 型，采用外控模拟量输入方式；
- 要求变频器的输出频率按照如图 5-40 所示的曲线变化；
- 用 PLC 程序控制变频器，并使用 EM235 模拟量模块输出模拟信号；
- 变频器的工作方式需手动设置（参数），工作过程由 PLC 进行控制；
- 需用组态王监控变频器的起/停，采用实时趋势曲线监测频率变化过程；
- PLC 的输入端也要接入一个开关，用来起/停变频器。

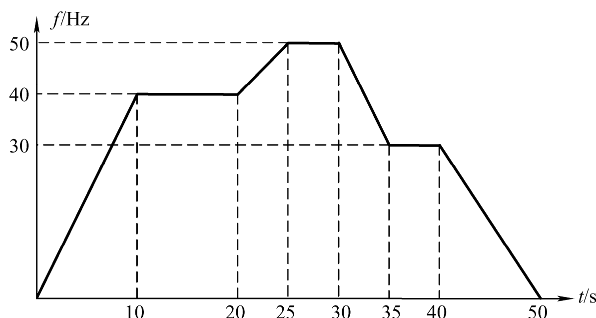


图 5-40　变频器输出频率变化曲线

4. 变频器参数设置

P0003=2	用户的参数访问级，2 为扩展级
P0010=0	0 为运行；1 为快速调试；30 为初始化
P0100=0	地区工频选择，0 为 50Hz
P0304=220	电动机额定电压为 220V
P0307=0.18	电动机额定功率为 0.18kW
P0310=50	电动机额定频率为 50Hz
P0311=2800	电动机额定速度为 2800 转/分钟
P0700=2	由端子排输入起/停命令（外控）
P0701=1	数字量输入端口 1 的功能，1 为接通正转
P1000=2	选择频率设定值，2 为模拟量输入
P1080=0	最小频率为 0Hz
P1082=50	最大频率为 50Hz
P3900=1	快速调试结束

5. 接线、线路端口作用及操作步骤

变频器调速控制系统原理图如图 5-41 所示，按图把线接好。图中变频器是 MM420 型的，属于单相输入/三相输出那种，除了 5 根主线路外，还有控制线路，控制线路又分输入信号与输出信号，本题没有利用输出信号又篇幅所限，所以都没画出来，输入信号也没画全，用哪个画哪个。8 号端子是自身直流 24V 电源的正极，可作为数字量输入信号的公共端；5 号端子是数字量输入信号的 1 号端子通常又称 DIN1，属于多功能端子，这端子信号起什么作用由用户自己选，用参数 P0701 来定，本题定为 1 就是此端有信号，变频器要正转的；3 号端子是模拟量输入信号的正端；4 号端子是模拟量输入信号的负端，二端间输入的是直流电压信号 0～10V。再看看 EM235，它是 PLC 的扩展模块，专门用来模拟量往来。本题没有模拟量输入信号，如图把输入端口都各自短接即可，唯一的一个模拟量电压型输出端口 M0～V0 是必须得用的，由它给变频器 3～4 端送可变的电压信号。

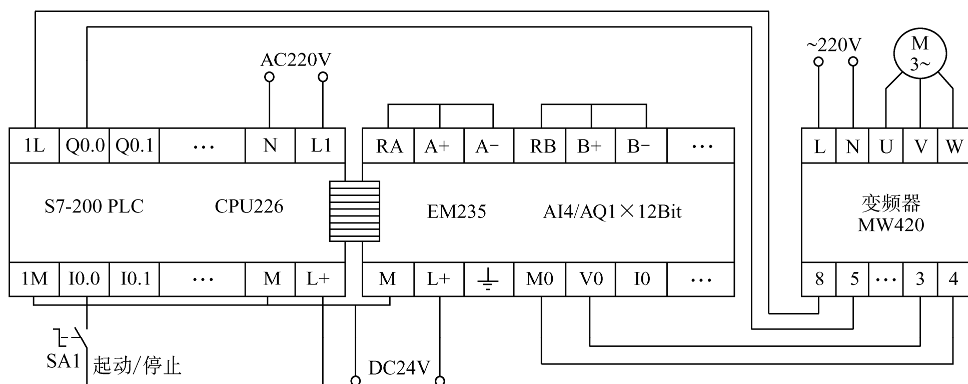


图 5-41 变频器调速控制系统原理图

接好线后，通上电，把编写好的程序传给 PLC，模式开关定为“RUN”，然后把编程软件 STEP7 退出，因为还有组态王监控画面要与 PLC 通信，所以通信口不能占着。变频器的参数按本题的要求设置到位，接下来就是设置组态王监控画面，具体步骤后叙。按要求是即可以在 PLC 上用硬件开关控制系统的起/停，也可在计算机显示屏上用制作好的“软开关”控制起/停。起动后，变频器就应按照图 5-40 所给定的频率变化曲线输出频率，从而去控制电动机速度，在组态王画面上会同步出现输出频率的变化曲线，这就是本题想要达到的监控目的。

6. PLC 程序及编程思路

PLC 程序如图 5-42 所示，图示为梯形图。组态王监控画面上所作的按钮、开关如作为 PLC 的输入信号要用 PLC 的辅助继电器 M（bit），如程序中的 M1.0。本题变频器的控制特点是属于外控模拟量输入形式，让 PLC 的 Q0.0 给变频器的数字量输入信号第一端提供一个连续的信号，作为变频器的方向信号。因为是模拟

量信号而不是数字量信号（段速），所以上坡与下坡的斜率要按控制要求的曲线走，参见图 5-40，这样需用 PLC 的程序控制自己的模拟量输出端口，使其发给变频器模拟量输入端口的数据（电压）严格按照曲线走，从而达到调速的目的。

图 5-42　PLC 梯形图程序

7. 组态王监控画面制作

① 为工程项目定文件夹及名称　打开组态王软件，第一个界面如图 5-43 所

示，如果想打开已有文件，会在“工程名称”栏下出现已设计好的工程项目，双击打开即可。如想新建，就单击“新建”按钮，这时出现的就是图 5-43，看一看，我们不需要做什么，直接单击“下一步”按钮，出现如图 5-44 所示界面。给新建工程定文件夹，定好后点“下一步”按钮，出现如图 5-45 所示界面。定新建工程名称，定好后单击“完成”按钮，出现如图 5-46 所示界面。蓝色光标所指就是当

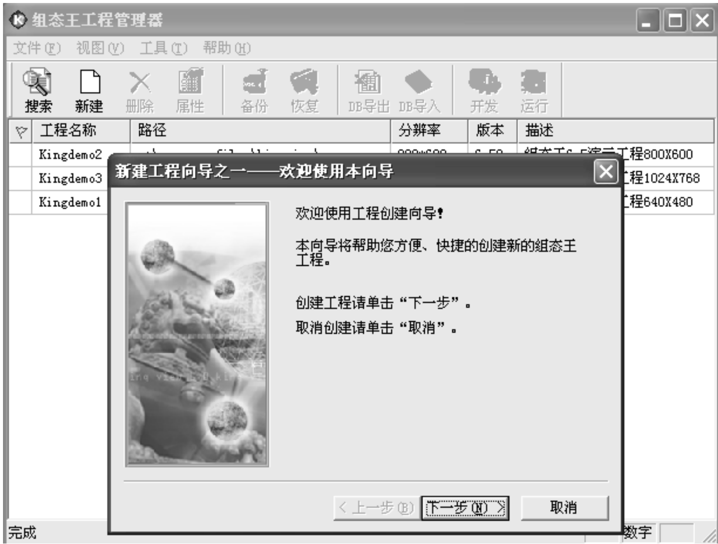


图 5-43　新建一个工程项目的第一步

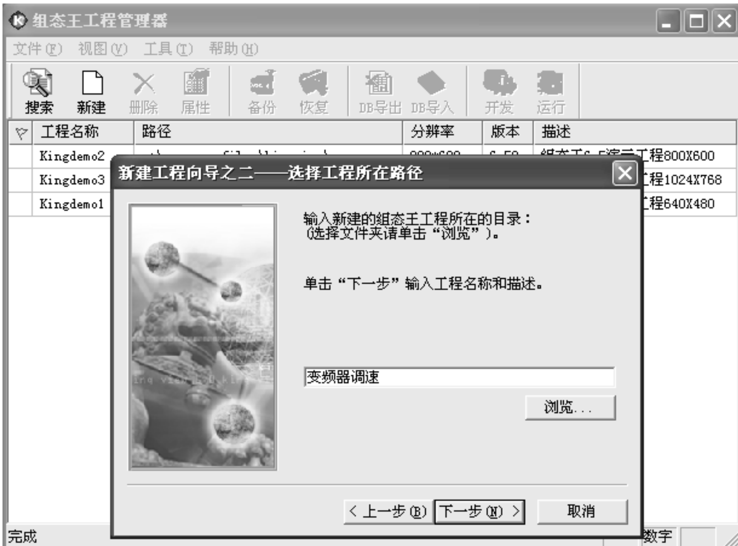


图 5-44　为新建工程定文件夹



图 5-45　为新建工程起名字



图 5-46　新建的工程项目已在工程名称栏下

前工程，双击它就会出现如图 5-47 所示该工程的浏览器界面。

② 设置通信参数　在工程目录显示区双击 COM1，开始定“通信对象”，也就是定监控画面将与什么智能设备进行通信，出现如图 5-48 所示界面。在这里定与通信有关的参数，没有什么可改动的，也就是波特率用户可改一下，然后单击“确定”按钮，出现如图 5-49 所示界面。这时双击工程目录内容显示区的那个图标，出现如图 5-50 所示界面。定通信设备的类别，如选择 PLC，就双击 PLC，弹

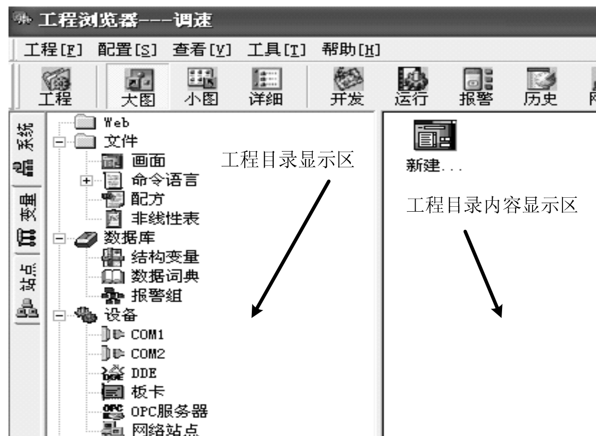


图 5-47　开始为通信设备定通信参数



图 5-48　定通信速率及数据长度等

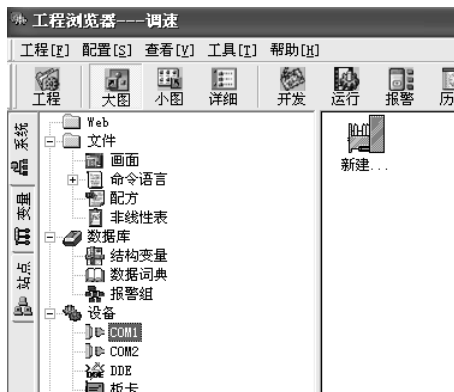


图 5-49　定通信设备类别

出可与组态王监控画面通信的所有 PLC 的名称，选择西门子，弹出该品牌的各种规格，再选择 S7-200，出现如图 5-51 所示界面。定通信方式，也就是上位机与 PLC 之间的通信方式，应该是 PPI，单击“下一步”按钮，出现如图 5-52 所示界面。这一步是给通信设备起名字，起什么名字都可以，如无所谓也可直接单击“下一步”按钮，这时默认的设备逻辑名是“新 IO 设备”，下一步如图 5-53 所示，定



图 5-50　定 PLC 的品牌及规格



图 5-51　设置通信方式



图 5-52 为通信设备起名字

串行通信口，即“COM”口，如何知道 PC 与 PLC 通信时 PC 这边用的是哪个串口，通常都是从 PLC 的编程软件 STEP 7 中获取，找到后直接点击那个口即可，点“下一步”，出现如图 5-54 所示界面。定 PLC 的地址，这仍然要从 STEP 7 软件中获取，单台 PLC 的默认地址是 2，如不是单台或先前做过设置，按设置的写，



图 5-53 选择计算机这边的串行通信口

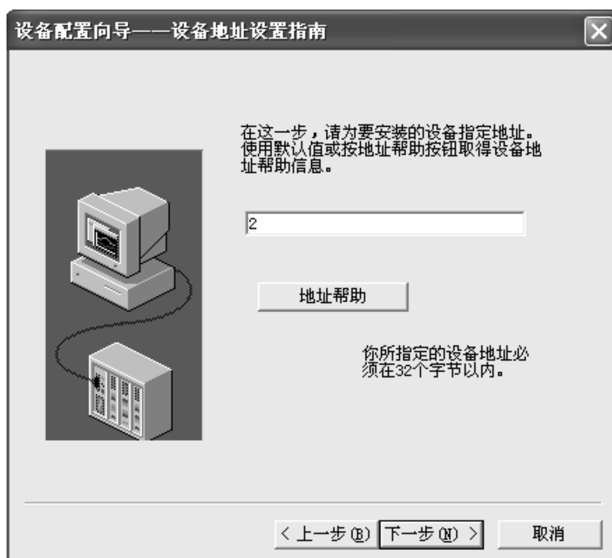


图 5-54　定 PLC 的地址

这一点设计者自己是清楚的。再单击“下一步”按钮，出现如图 5-55 所示界面。到此通信设备的信息就定完了，如觉没错，就单击“完成”按钮。



图 5-55　通信参数设置完成

③ 定义监控画面中的变量　接下来是制作监控画面，在工程目录显示区点击“数据词典”，出现如图 5-56 所示界面。为将要制作的画面上出现的图素（画面

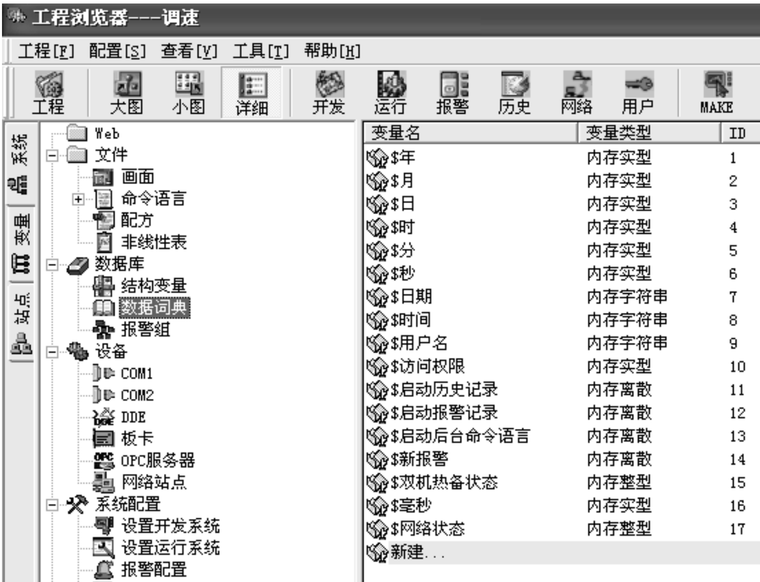


图 5-56　为画面上的图素定变量名

上的可动元素，像按钮、电动机、机械手等）起名字、定联系地址等。按本题要求，需要定一个按钮、一台电动机、一条反应速度变化的曲线，双击工程目录内容显示区的“新建”出现如图 5-57 所示界面。注：空白栏内的内容都是填写的，



图 5-57　为画面上的按钮定相关参数

在变量名栏写“按钮”；在变量类型栏点下拉箭头，定“I/O 离散”，意思是此变量是个数字量（闭合为 1，断开为 0），故称为“离散”，又因为此变量要送出去给外面设备作为 I 或者 O，所以称为“I/O”；在连接设备栏点下拉箭头，只会出现“新 I/O 设备”；在寄存器栏定“M1.0”，这一位将代表画面上的按钮出现在 PLC 程序中（参见图 5-42）；数据类型肯定是“Bit”（位）；读写属性定“读写”；采集频率定“100 毫秒”，这一变量就定完了，单击“确定”按钮继续定其他变量。接下来定电动机，过程与定按钮时完全相同，只是变量名为“电动机”，寄存器为“Q0.0”，如图 5-58 所示，监控画面上的电动机应与 PLC 程序中的 Q0.0 位是同步的，定好后单击“确定”按钮。最后为电动机速度变化曲线定义，还是双击工程目录内容显示区的“新建”（参见图 5-56），在弹出界面的选择栏内填上相关内容，如图 5-59 所示。在变量名栏写“变频器输出频率曲线”，在变量类型栏写“I/O 整数”，这是因为曲线应能反映出连续变化过程，与之对应的变量是 PLC 内能反映数据多少的区域而不是某个位，在寄存器栏选 V10，在数据类型栏选“SHORT”（字），实际上就是 VW10，一个字能表示的最大十进制数是 32767，所以最大值及最大原始值都填入此数，定义并没有结束，在此界面的最上一行单击“记录和安全区”，弹出如图 5-60 所示界面，在此单击“数据变化记录”，然后单击“确定”按钮，这样，画面上将要制作的 3 个变量都定义完了。



图 5-58　为画面上的电动机定相关参数

④ 编写命令语言　命令语言是一种类似 C 语言的程序，编写这种程序的目的



图 5-59　为画面上的曲线定相关参数

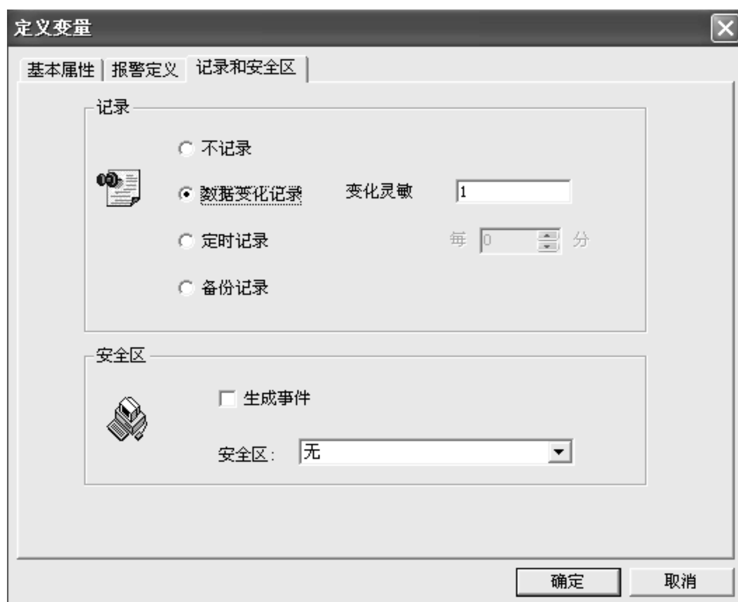


图 5-60　为曲线选择数据变化记录

就是让画面上的图素按控制要求动作，即使画面与现场设备没有通信，画面本身的图素之间也会按照“命令”动起来。比如，制作一个机械手抓取重物的画面，把 PLC 程序编写到位，把画面这里的“命令”编写到位，正式监控工作时，现场

的机械手在干什么，画面上的机械手也在干什么，给人的感觉是画面在监控现场，实际上是没有关系的，画面上的动作是通过“命令”制作出来的。假设现场的机械手出事故卡住了，送回画面一个信号，画面上的机械手也停止不动就可以了。要把二者统一起来，像是在同步工作，达到监控的目的，这就是设计者的任务。

组态王的命令语言中常用的是应用程序命令语言，以本题为例，简介应用程序命令语言的编写过程。在本题中，按要求是当开关拨动后，PLC 的输出 Q0.0 要闭合，直到开关断开（参见图 5-42）。在画面上画一个带自锁的按钮，当做开关，画一台电动机与 PLC 那里的 Q0.0 同步动作，动画制作后叙，这里只介绍实现这一动作过程的命令语言如何编写。在工程目录显示区（参见图 5-47）双击“命令语言”，在其栏下会出现 5 种语言，单击第一个“应用程序命令语言”会形成如图 5-61 所示界面，按照提示，在工程目录内容显示区双击那个图标，在弹出界面的编程区内，编写本题的程序，只有四行，如图 5-62 所示。程序简单明了，编写时要注意这几点：先写条件后写结果；所使用的各种字母及符号，一定要用界面栏下方所给的，不能在键盘上输入，否则无效；写条件时要带括号，括号内要用双等于号；变量一定是先前定义过的，否则无效，写完觉得可以了就单击“确定”按钮，这儿有语法自检功能，如有错误，会弹出提示，及时纠正，直到界面能退出，就说明命令语言编写完毕。逻辑正确与否只待画面运行时验证。

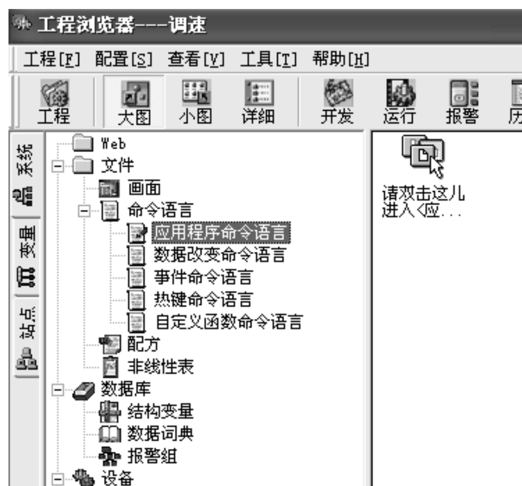


图 5-61　选择应用程序命令语言

⑤ 制作画面　在工程浏览器界面的工程目录显示区（参见图 5-47），单击“画面”，在工程目录内容显示区会出现一个新建图标，双击此标会弹出如图 5-63 所示界面，写入画面名称，单击“确定”按钮后会弹出“开发系统”界面，如图 5-64 所示。单击“图库”后会出现下拉菜单，选“打开图库”，在弹出的界面上单击“按



图 5-62　编写命令语言



图 5-63　建立画面

钮”，如图 5-65 所示。选中一个合适的按钮，双击它后，图库管理器就退掉了，在绘图界面上（开发系统）放置这个按钮，大小形状可随意调整，然后再打开图库，点击“马达”后出现各式电动机，同样选中一个合适的拖到开发系统界面上，再调整一下大小，此时的界面如图 5-66 所示。到此，本题需制作的 3 个图素已画



图 5-64　绘制图素

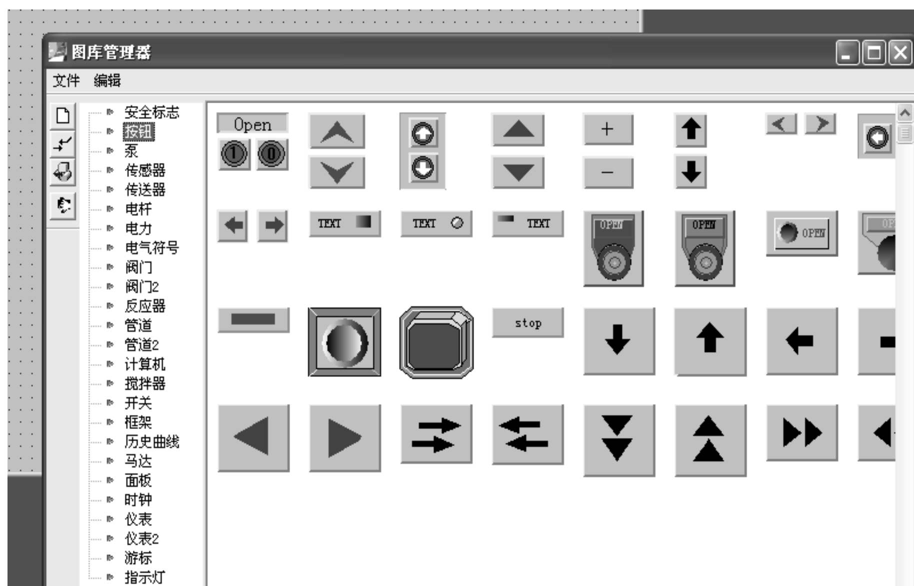


图 5-65　选择按钮

好 2 个，下面制作曲线。在开发系统界面点“工具”后出现下拉菜单，选中“实时趋势曲线”，如图 5-67 所示。点击后拖到开发系统界面的适当位置上，调整大小形状，形成如图 5-68 所示界面。这时，3 个图素就都制作完了，接下来是让图素如何“动”起来。

⑥ 动画连接　所谓动画连接，就是让图素按照其应起的作用与功能而动起来，模拟其物理存在去控制现场设备或受控于现场设备。在图 5-66 中，双击“按钮”图形，弹出“按钮向导”对话框，如图 5-69 所示，只需在变量名处写入此图形所代表的变量名或点旁边的问号，从下拉菜单中选取应代表的变量，其他各项根据要求选定，如不需动什么，单击“确定”按钮，按钮的动画连接就已作好。

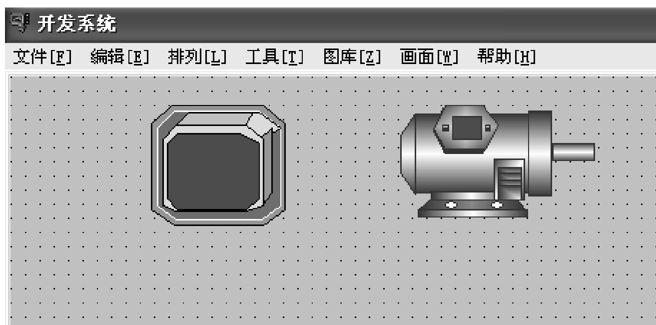


图 5-66 制作好的 2 个图素



图 5-67 制作速度变化曲线

电动机的动画连接与按钮的相同，下面制作曲线的动画连接。双击曲线弹出如图 5-70 所示界面，在曲线 1 表达式的空白框里填写所代表的变量名或点旁边的问号，从下拉菜单中选取应代表的变量，其他各项根据要求选定，然后点击本界面的另一层“标识定义”，如图 5-71 所示，在此定时间轴全程显示的长度，本题全程需时 50s，所以定 60s 足够了，横坐标的时间轴定完了，纵坐标是以百分比的形式出现，最高是输出信号的 100%，在这不用改什么，与这条曲线相对应的是 PLC 的 VW10 变量存储器，一个字是 16 位，能表示的最大十进制数是 32767，所以 100% 代表的是 32767，根据实验室实验测得变频器输出 10Hz 时，与之对应的 VW10 中

的数据是 6500 左右，以此推算，20Hz 时 13000；50Hz 时是 32500。这样，将时间定好其他不用动了，单击“确定”按钮后，弹出的画面如图 5-72 所示的曲线部分，曲线的动画连接也做完了，至此，3 个图素的动画连接全都定好。

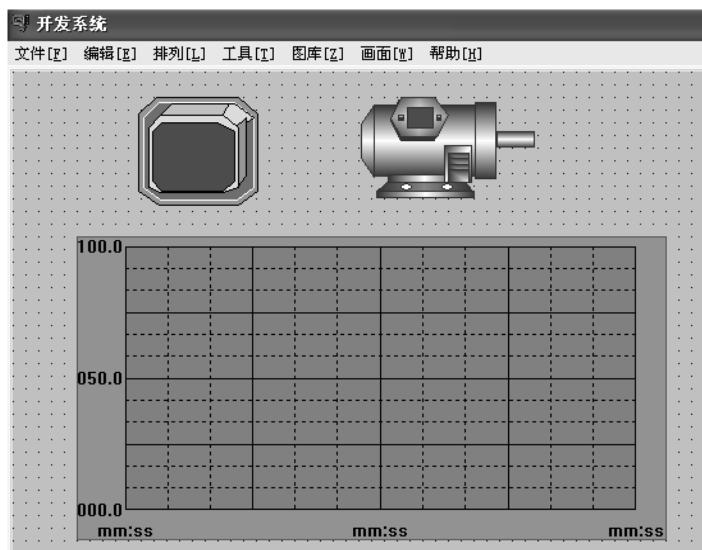


图 5-68　已制作成的 3 个图素的静态画面

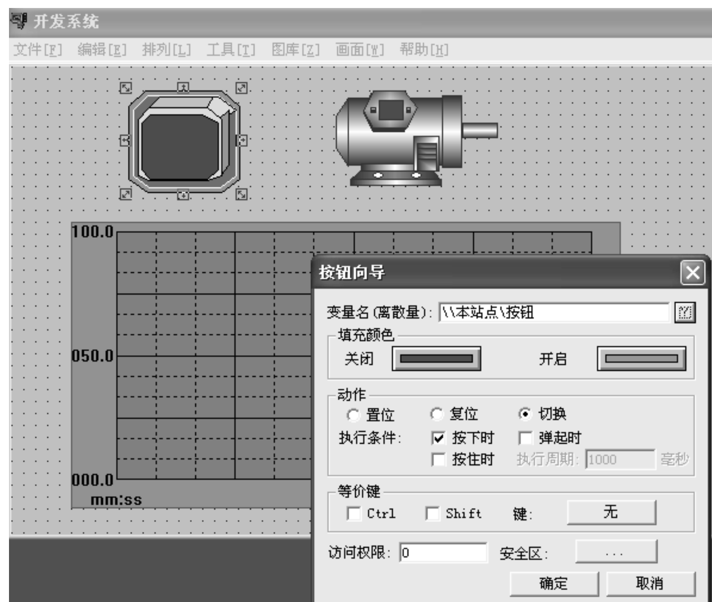


图 5-69　为按钮设置动画连接

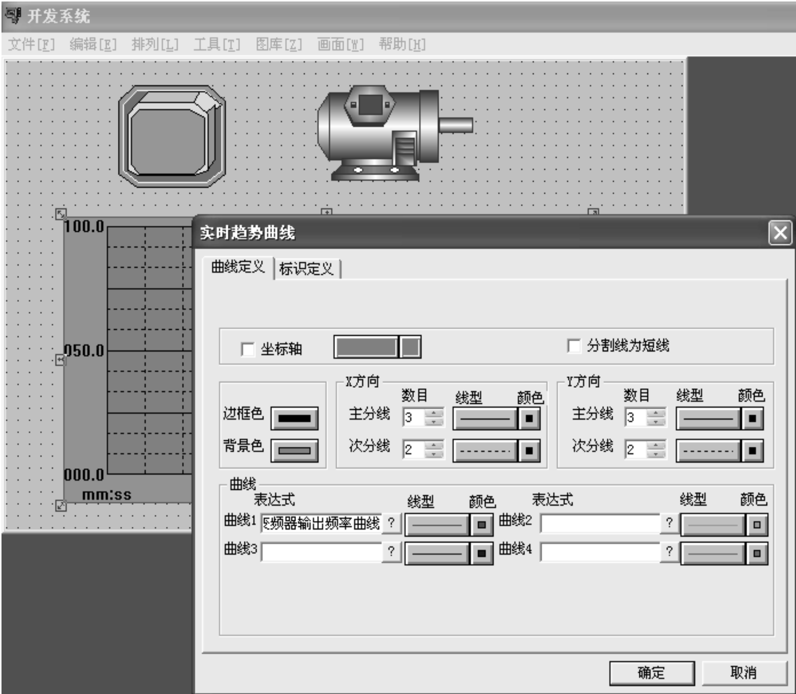


图 5-70 为曲线设置动画连接

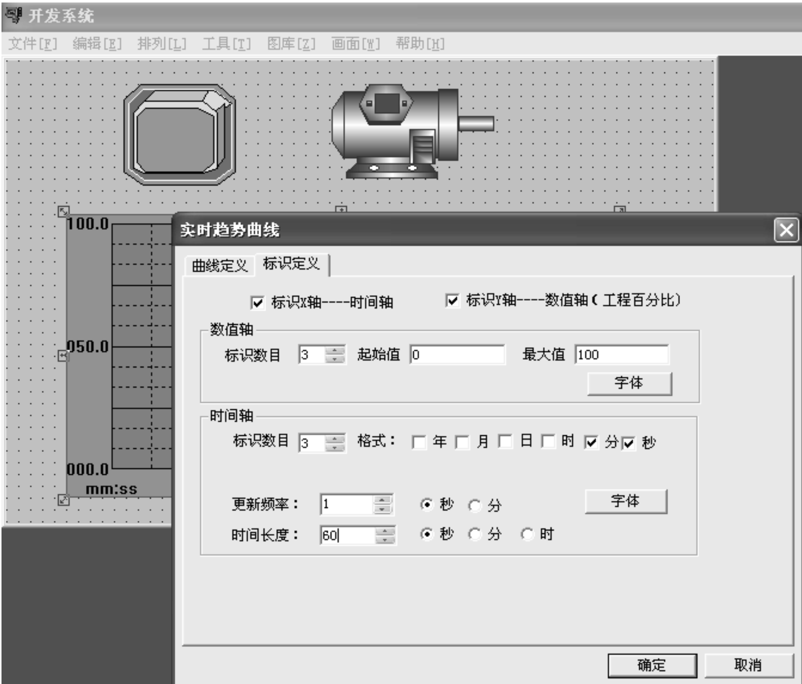


图 5-71 为曲线设置标识

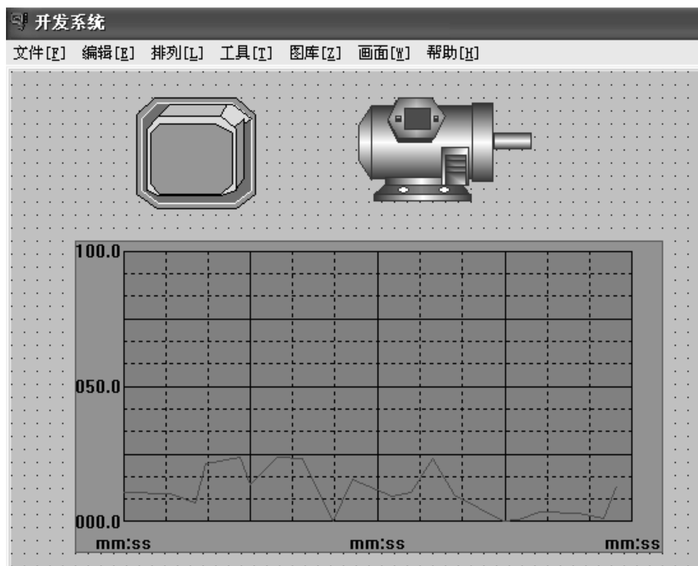


图 5-72　3 个图素的动画连接制作完毕

⑦ 保存、切入运行（监控）画面　都设置好了，该投入工作了，这时一定要做的事情是保存，在图 5-72 中单击“文件”会出现下拉菜单，如图 5-73 所示，选“全部存”，这一步做完，还单击“文件”，在下拉菜单中选“切换到 View”，也就是转入运行系统，单击后，界面转换成如图 5-74 所示，整个界面一片空白，这时单击“画面”，在出现的下拉菜单中选“打开”，弹出一个“打开画面”的对话

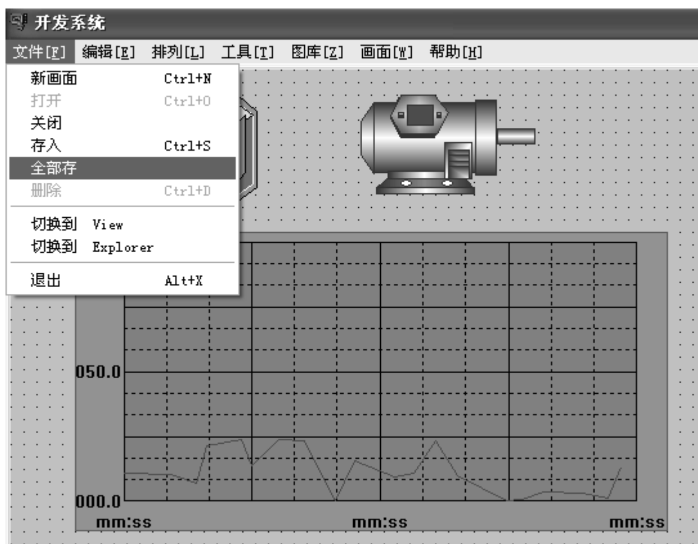


图 5-73　保存制作的画面



图 5-74　运行系统的切换过程

框，图 5-74 中右侧图，画面名称处显示的就是本题的画面名称，点击并确定后，画面就成为运行系统了。

给现场的变频器通上电，参数设置好；PLC 通上电且程序早已载入，将模式开关拨到“RUN”，这时，点击画面上的按钮，系统就可运行了，电动机并不能转，由指示灯指示其工作状态，曲线按控制要求显示电动机的速度变化过程，每隔 1s 曲线由右向左移动一次，图 5-75 所示为本题要求的监控画面。

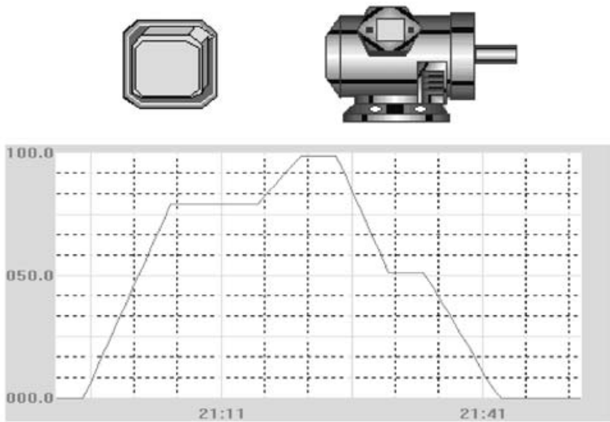


图 5-75　变频器调速过程的监控画面

例十二、车间生产流水线产品运输控制程序

1. 任务描述

在生产车间，经常会遇见一台装料车在自动化生产线上，根据请求地点信号要求进行多地点随机装料，收集成品。

现某食品生产车间有四个工作台和一个成品库，直线排列。操作工将食品包装成袋，然后呼叫小车把包装袋送到成品库，小车往返于各个工作台之间，根据请求在某个工作台限时限量装袋，当小车装满 60 袋时，就自动开往成品库卸袋。待卸袋完毕，然后在根据请求开往某站继续工作，由此往返。如图 5-76 所示。

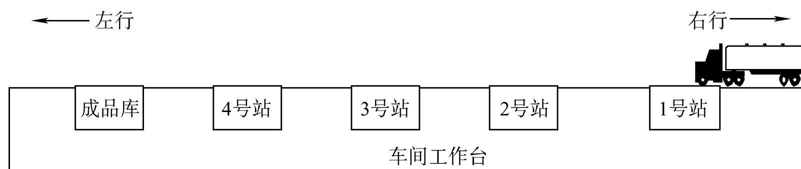


图 5-76　生产车间加工现场图示

2. 控制要求

（1）每个工作台都有一个呼叫按钮。当需要小车过来装袋时，按一下按钮，系统接到呼叫信号就登记下来，同时本站呼叫记忆灯点亮，说明信号已接到。小车就会根据自己的忙闲程度决定什么时候来到。

（2）装袋时有两个要求，一是每次最多装 10 袋，二是每次最长停留 8s，也就是说不到 8s 就把 10 袋都装完了，这时如有其他呼叫等待，小车就离开这里。如果用尽 8s 却装不到 10 袋，此时若已有呼叫信号，小车也照样离开这里。目的是避免压车。

（3）如 10 袋已装完，8s 时间也到了仍没有别的工作台呼叫。本台可以继续装料，但一旦有别的工作台呼叫信号，小车立即出发离开这里。

（4）数码管显示小车所处的站台。

（5）只要车上装满 60 袋，对所有呼叫信号都不应答只保留登记顺序，小车直接开往成品库。待卸料完毕后重新按登记顺序继续应接。

（6）用一个按钮起动程序，另一个按钮停止程序。

（7）工作台之间的距离，停靠位置由旋转编码器决定。旋转编码器每发出 1000 个码就是一个工作台的位置。这样 4 个工作台加一个成品库，共需 4000 个码。第一个工作台应设基准位置开关（传感器）。

（8）小车到成品库卸袋，共需 10s，时间到后就认为是卸料完毕。

3. 程序设计

（1）主要部分电路设计说明　在程序设计中，采用高速计数器（HSC）指令，开通旋转编码器计数，用以计算小车的行走距离，实现准确停靠。因为是实验室模拟运行，所以本程序没有考虑小车停靠时的减速段，也就是直接起，直接停，使用的是直流 24V 的小型电动机。

高速计数器有四种基本类型，本程序中选用的类型为 A/B 相正交计数器。

程序中除了利用高速计数器计算距离是个难点，还有就是交通信号的建立、

响应、截车、停靠及定时定量装料。按要求，小车停靠的位置还须有七段数码管显示。

（2）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配。车间生产流水线产品运输 PLC 接线图如图 5-77 所示，系统的 I/O 说明见表 5-4。

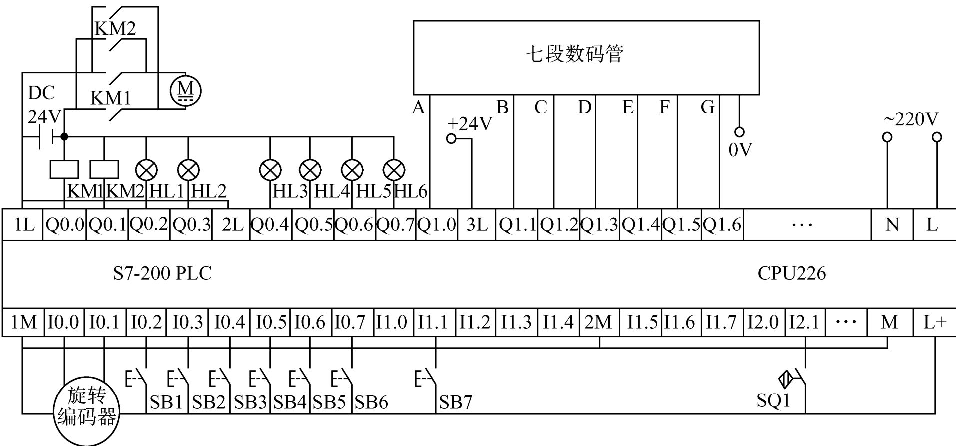


图 5-77 车间生产流水线产品运输 PLC 接线图

表 5-4 系统的 I/O 说明

I/O	用　途	I/O	用　途
I0.0、I0.1	高速计数器	Q0.6	3 号站呼叫灯（HL5）
I0.2	停止按钮（SB1）	Q0.7	4 号站呼叫灯（HL6）
I0.3	开始按钮（SB2）	Q1.0～Q1.6	数码管显示
I0.4	1 号站呼叫按钮（SB3）	M0.0	开始按钮按下，得电并自锁
I0.5	2 号站呼叫按钮（SB4）	M0.1	1 号站呼叫时，得电自锁
I0.6	3 号站呼叫按钮（SB5）	SM0.0	常 ON 状态
I0.7	4 号站呼叫按钮（SB6）	M0.2	2 号站呼叫时，得电自锁
I1.1	装料按钮（SB7）	M0.3	3 号站呼叫时，得电自锁
I2.1	传感器（SQ1）	M0.4	4 号站呼叫时，得电自锁
Q0.0	反转（KM1）	M0.5	每到一站，得电并接通时间继电器
Q0.1	正转（KM2）	M0.6	2 号站到时得电并截车
Q0.2	反转小信号灯（HL1）	M0.7	3 号站到时得电并截车
Q0.3	正转小信号灯（HL2）	M1.1	1 号站到时得电并截车
Q0.4	1 号站呼叫灯（HL3）	M1.2	4 号站到时得电并截车
Q0.5	2 号站呼叫灯（HL4）	M1.3	60 件装料完成，得电并直接送至成品库

（续）

I/O	用　　途	I/O	用　　途
M1.4	小车到达成品库得电，起动计时器 T38	M3.0	到达 3 号站时得电并断开 3 号站呼叫灯
M1.5	T38 得电后自锁	M4.0	到达 4 号站时得电并断开 4 号站呼叫灯
M2.1	小车在 1 号站，有呼叫时得电	M5.0	8s 时间到或装满 10 件，得电反转
M2.2	小车在 4 号站或成品库，有呼叫时得电	M6.0	8s 时间到或装满 10 件，得电正转
M1.0	到达 1 号站时得电并断开 1 号站呼叫灯	SMB37	监控高数计数器 HSC0
M2.0	到达 2 号站时得电并断开 2 号站呼叫灯		

（3）控制程序的梯形图及注释如图 5-78 所示。

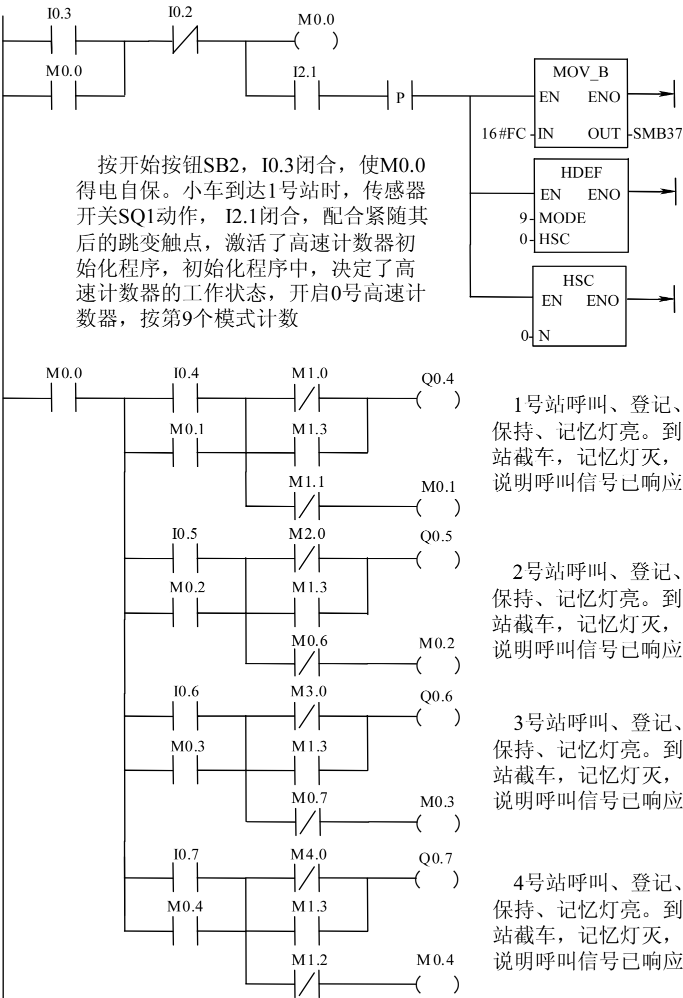


图 5-78　控制程序的梯形图及注释

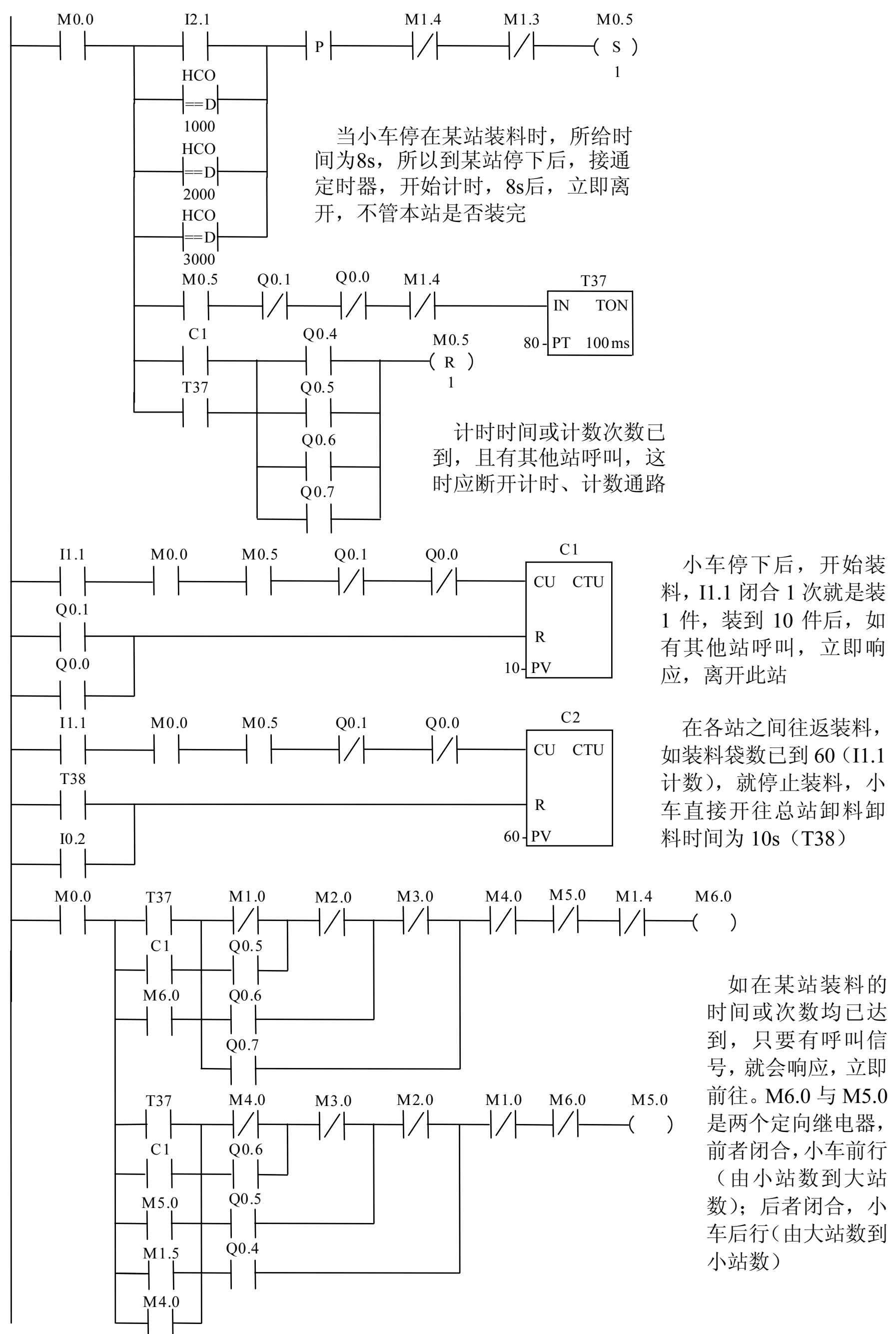


图 5-78　控制程序的梯形图及注释（续）

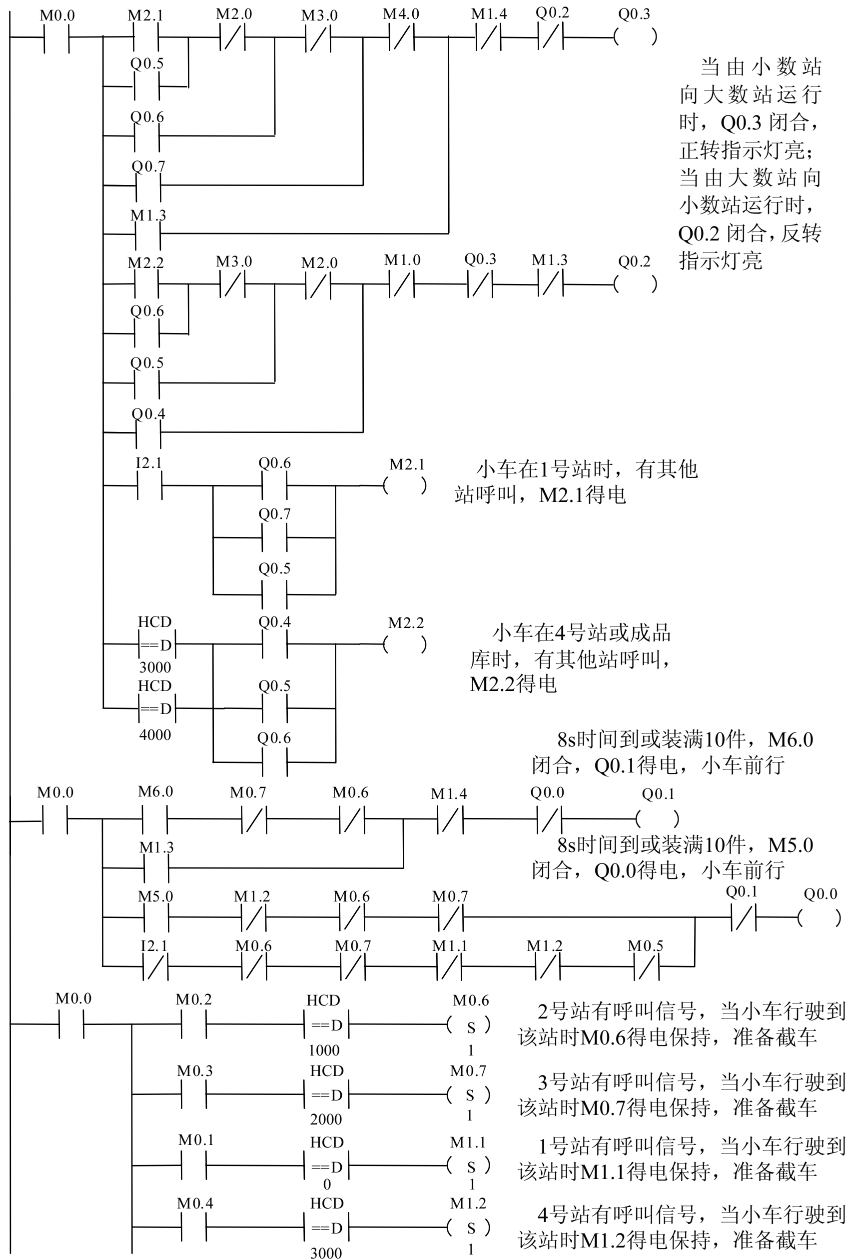


图 5-78　控制程序的梯形图及注释（续）

定时定量装料设定值已到，或车上已装满 60 件，此刻将复位所有呼叫信号，不再应答。小车直接开往成品库

60 件装料完成，M1.3 得电，使小车直接开往成品库

小车到达成品库，M1.4 得电，T38 起动定时器，同时使 M1.3 复位

小车到达成品库后，等 10s 就相当于卸料完毕，同时数码管显示 5，就是成品库

一旦有呼叫信号，那肯定是Q0.0得电，使M1.4复位，同时小车往左行

到1号站，数码管显示1

到2号站，数码管显示2

到3号站，数码管显示3

到4号站，数码管显示4

按停止按钮时，复位所有中间继电器

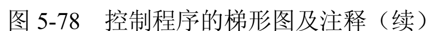


图 5-78　控制程序的梯形图及注释（续）

例十三、机械手搬运产品控制程序

1. 控制要求

（1）按下停止按钮，一切动作停止。

（2）有手动与自动两种状态，手动与自动状态可以切换。

（3）手动时，按下方向按钮，机械手向要求的方向运动。到了装料地点，按下电磁阀按钮，机械手抓紧。到了卸料地点，机械手松开，卸料。

（4）当转换到自动状态时，如机械手不在原点，机械手先复位，回到原点。然后按照路线运动。机械手运动路线：机械手臂左移→左移至指定位置后→抓手张开夹紧物品→机械手臂右移→右移至指定位置后→机械手臂上升→上升至指定位置后→机械手臂左移→左移至指定位置后→上层检测到有信号→抓手张开将物品放下→右移至指定位置后→机械手臂下降→下降至指定位置。至此，一个工作周期结束。其机械手工作过程示意图如图 5-79 所示，图中 SQ1～SQ4 为限位开关。

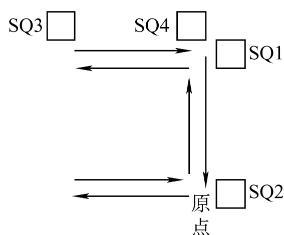


图 5-79　机械手运动示意图

2. 程序设计

（1）*X* 轴、*Y* 轴的运行需要两台步进电动机。脉冲输出（PLS）指令被用于控制在高速输出（Q0.0 和 Q0.1）中提供的“脉冲串输出”（PTO）和“脉宽调制”（PWM）功能。

（2）根据控制要求，首先要确定 I/O 个数，进行 I/O 分配。系统的 I/O 说明见表 5-5，机械手搬运产品 PLC 接线图如图 5-80 所示。

（3）控制程序梯形图如图 5-81 所示。

表 5-5 系统的 I/O 说明

I	用 途	O	用 途
I0.0	停止按钮（SB1）	Q0.0	纵轴脉冲输出（CP）
I0.1	手动按钮（SB2）	Q0.1	横轴脉冲输出（CP）
I0.2	自动按钮（SB3）	Q0.2	纵轴方向输出（DIR）
I0.3	上层限位开关（SQ1）	Q0.3	横轴方向输出（DIR）
I0.4	下层限位开关（SQ2）	Q0.5	电磁阀控制
I0.5	左层传感器（SQ3）	Q0.6	纵轴上行指示灯（HL1）
I0.6	右层传感器（SQ4）	Q0.7	纵轴下行指示灯（HL2）
I0.7	电磁阀控制（SB4）	Q1.0	横轴左移指示灯（HL3）
I1.0	上行按钮（SB5）	Q1.1	横轴右移指示灯（HL4）
I1.1	下行按钮（SB6）		
I1.2	左行按钮（SB7）		
I1.3	右行按钮（SB8）		

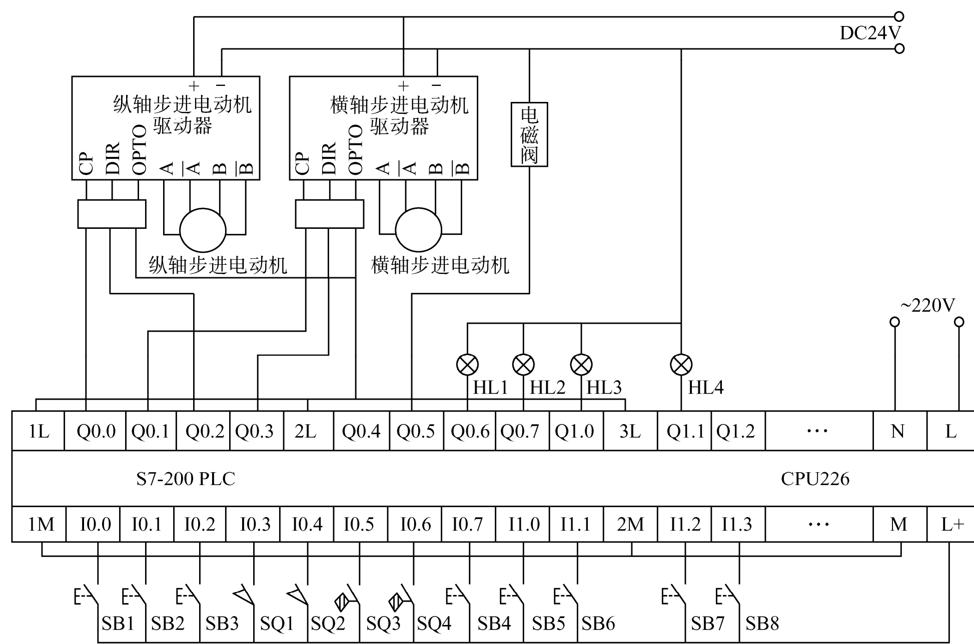


图 5-80 机械手搬运产品 PLC 接线图

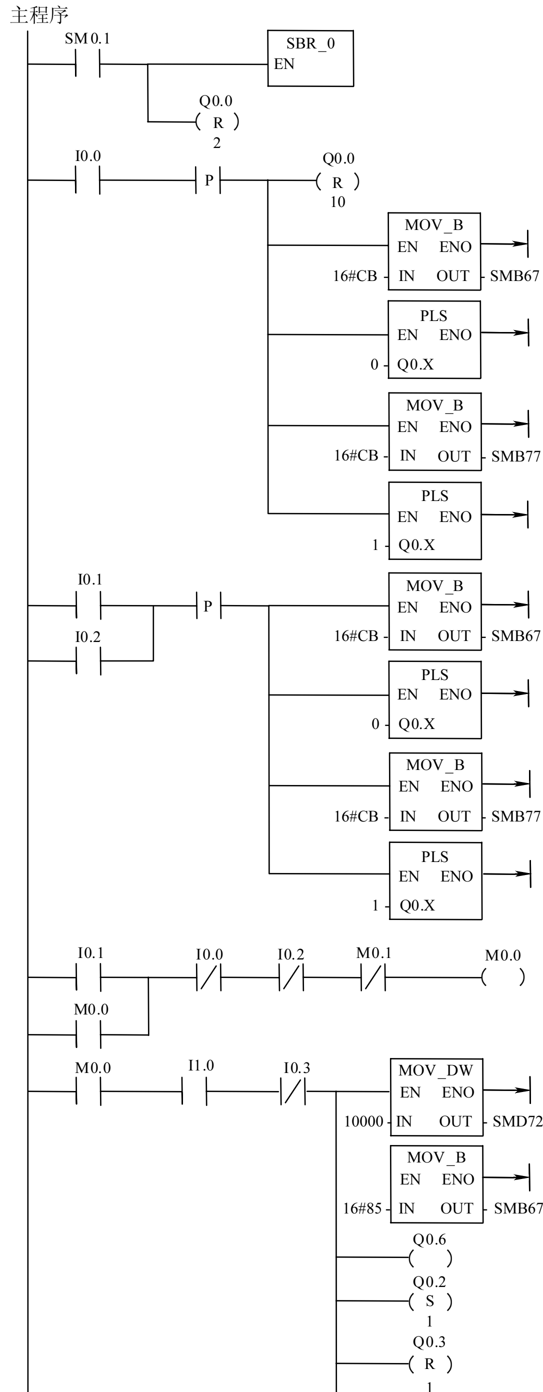


图 5-81 控制程序主程序梯形图

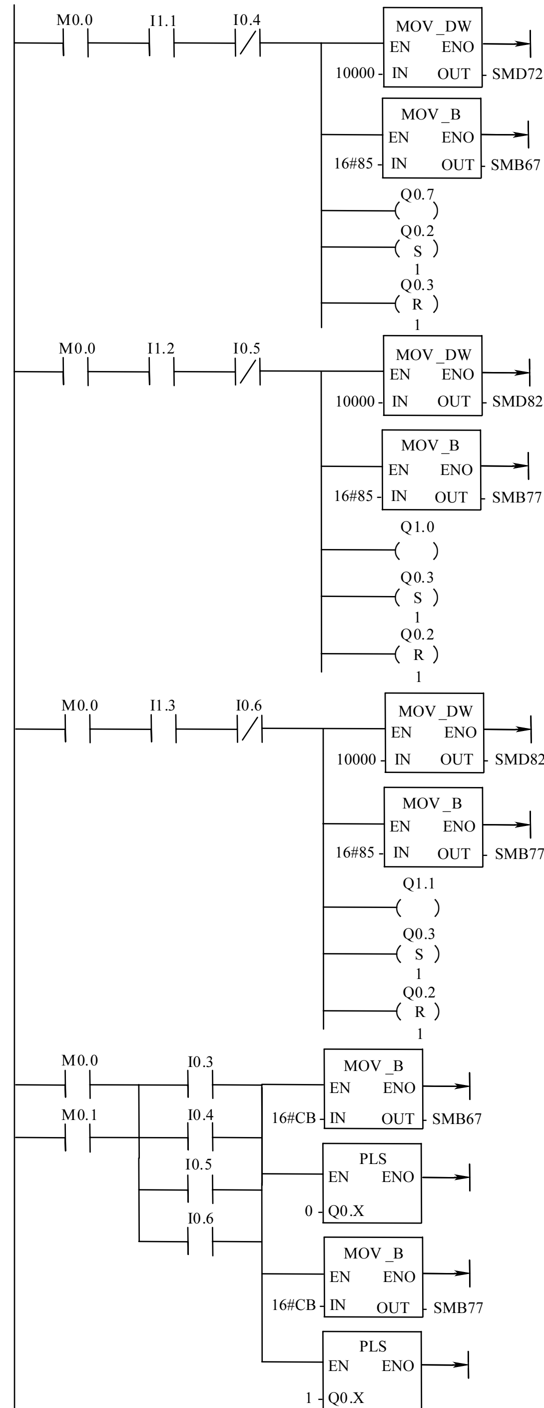


图 5-81　控制程序主程序梯形图（续）

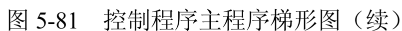


图 5-81　控制程序主程序梯形图（续）

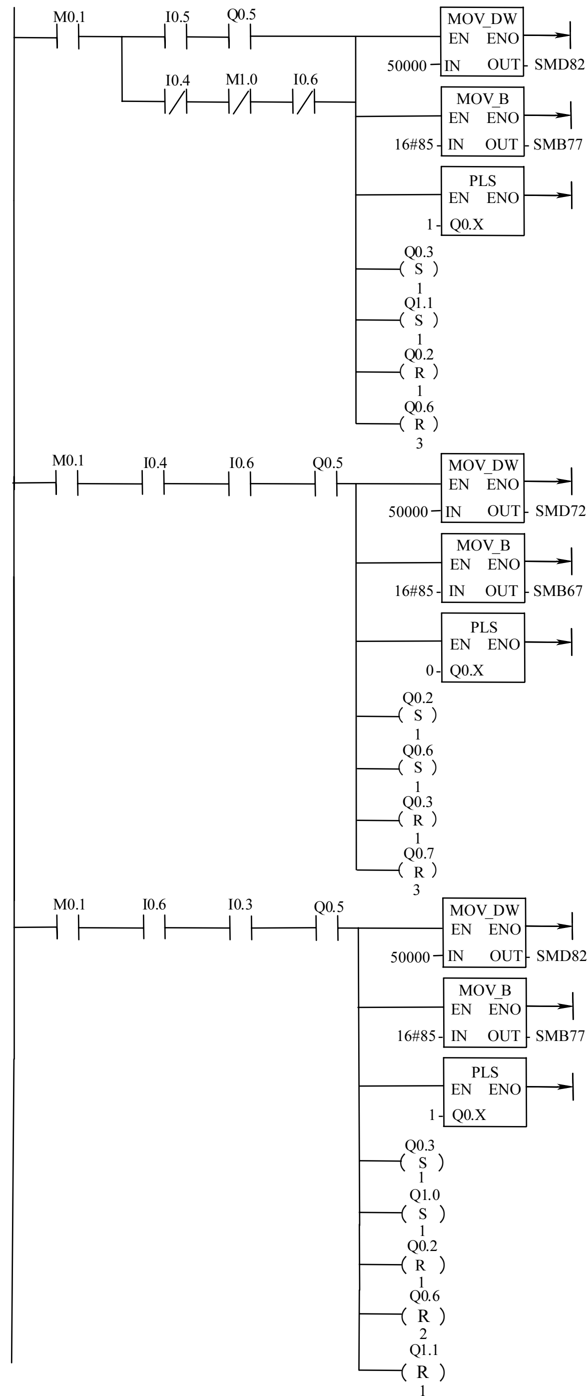


图 5-81 控制程序主程序梯形图（续）

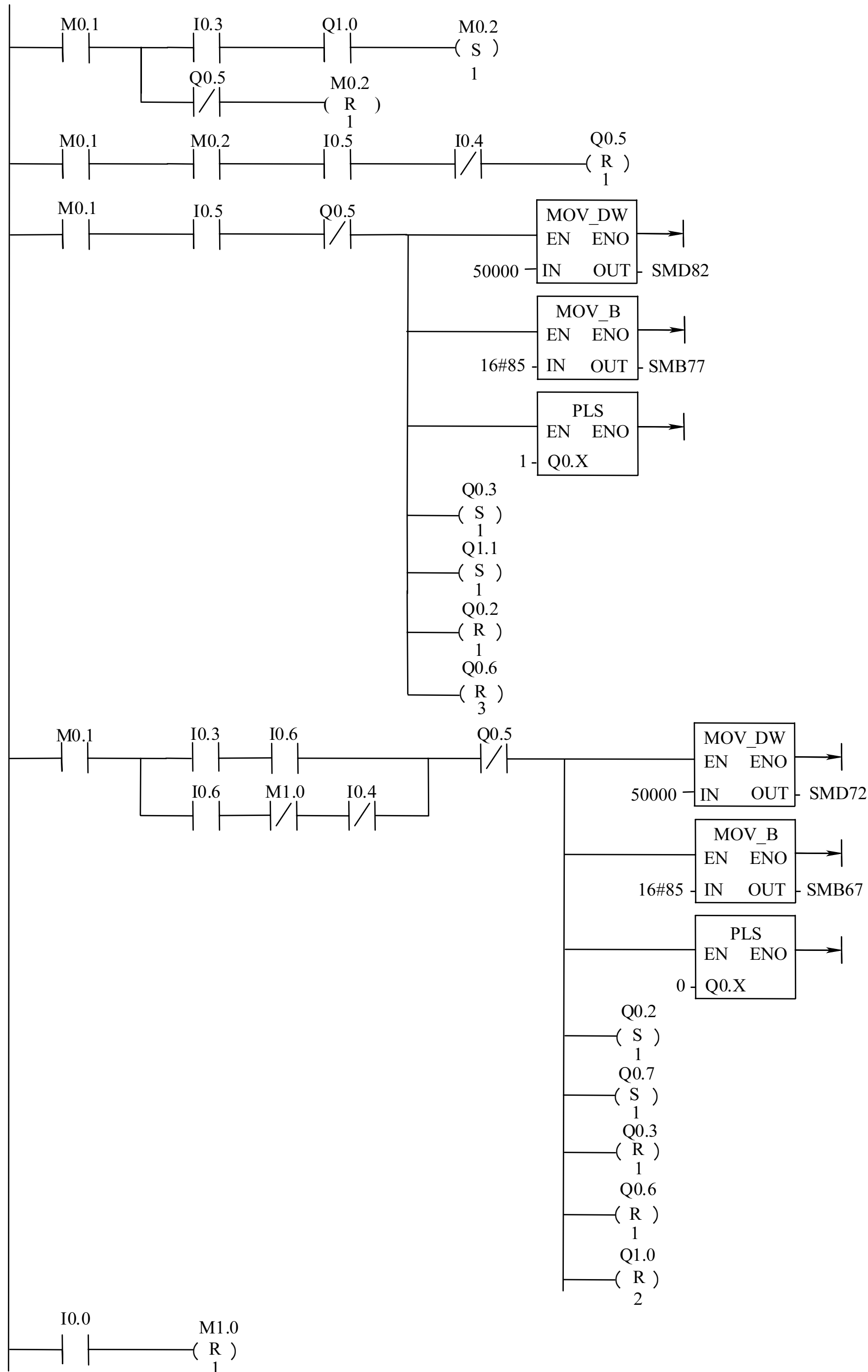


图 5-81　控制程序主程序梯形图（续）

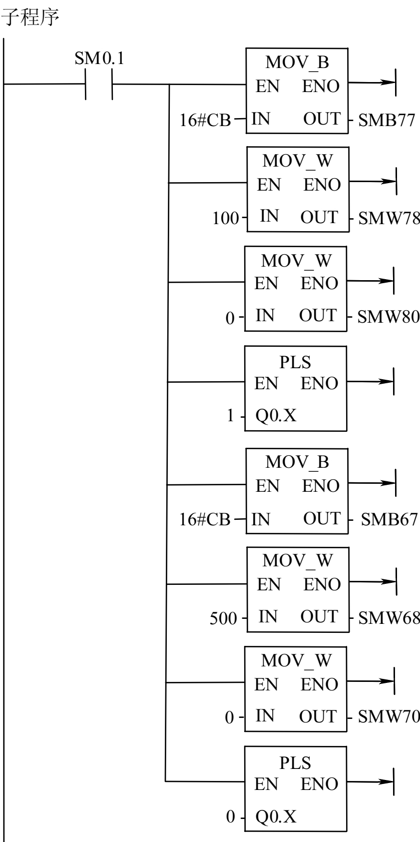


图 5-81 控制程序主程序梯形图（续）

第六章　STEP 7-Micro/WIN 编程软件

第一节　软件安装和设置

1. 简介与安装条件

S7-200 PLC 的编程软件是 STEP 7-Micro/WIN。在个人计算机 Windows 操作系统下运行，它的功能强大，使用方便，简单易学。CPU 通过 PC/PPI 电缆或插在计算机中的专用通信卡与计算机进行通信。此软件支持三种编程模式：LAD（梯形图）、FBD（功能模块）和 STL（语句表），便于用户选用，三种编程模式间可以相互转换。Micro/WIN 还提供程序在线编辑、调试、监控，以及 CPU 内部数据的监视、修改功能；支持符号表编辑和符号寻址，例如指定符号“电动机正转”对应于地址 Q0.0，使程序便于理解与寻找；支持子程序、中断程序的编辑，提供集成库程序功能，以及用户定义的库程序。

PLC 之间的网络通信、模拟量控制、高速计数器和 TD200 文本显示器的编程设计可以说是 S7-200 PLC 程序设计中的难点，STEP 7-Micro/WIN 为此设计了大量的向导，通过对话方式，用户只需要输入一些参数，就可以实现参数设置，自动生成用户程序。用户还可以通过系统块来完成大量的参数设置。

STEP 7-Micro/WIN 需要安装、运行在使用 Microsoft（微软）公司的 Windows 操作系统的计算机上。STEP 7-Micro/WIN V4.0 可以在 Microsoft 公司出品的如下操作系统环境下安装：

Windows 2000，SP3 以上。

Windows XP Home。

Windows XP Professional。

对计算机的硬件有如下要求：

任何能够运行上述操作系统的 PC 或 PG（西门子编程器）。

至少 350MB 硬盘空间。

Windows 操作系统支持的鼠标。

2. 安装

第一步：关闭所有应用程序，包括 Microsoft Office 快捷工具栏，在光盘驱动器内插入安装光盘。如果没有禁止光盘插入自动运行，安装程序会自动运行，或者在 Windows 资源管理器中打开安装光盘上的“Setup.exe”文件。

第二步：按照安装程序的提示完成安装。

① 运行 Setup 程序，选择安装程序界面语言，默认使用英语，如图 6-1 所示。



图 6-1 选择安装程序界面语言

② 选择安装此软件的文件夹，如图 6-2 所示。

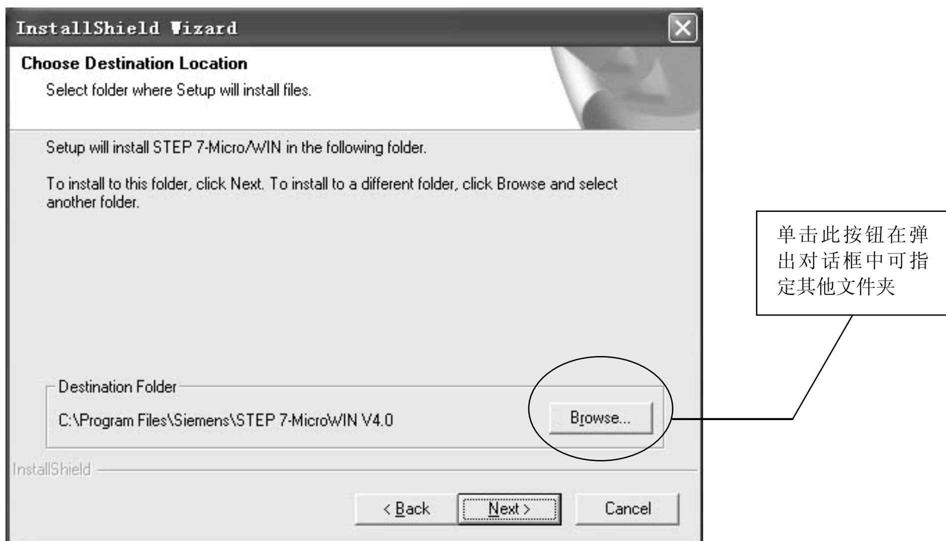


图 6-2 选择安装文件夹

③ 安装过程中，会出现“Set PG/PC Interface”窗口，如图 6-3 所示。这时让我们决定通信方式，确认“PC/PPI cable（PPI）”后，单击“OK”按钮，程序继续安装。

④ 安装完成后，弹出如图 6-4 所示的确认界面，单击对话框上的“Finish”（完成）按钮，重新启动计算机。

⑤ 重新启动后，用鼠标双击 Windows 桌面上的 STEP 7-Micro/WIN 图标，或者在 Windows 的“开始”菜单中找到相应的快捷方式，运行 STEP 7-Micro/WIN 编程软件，如图 6-5 所示。打开编程软件界面，使用“Help”→“About”的菜单命令查看软件版本信息，如图 6-6 所示。详细的版本信息如图 6-7 所示。



图 6-3　“Set PG/PC Interface”窗口

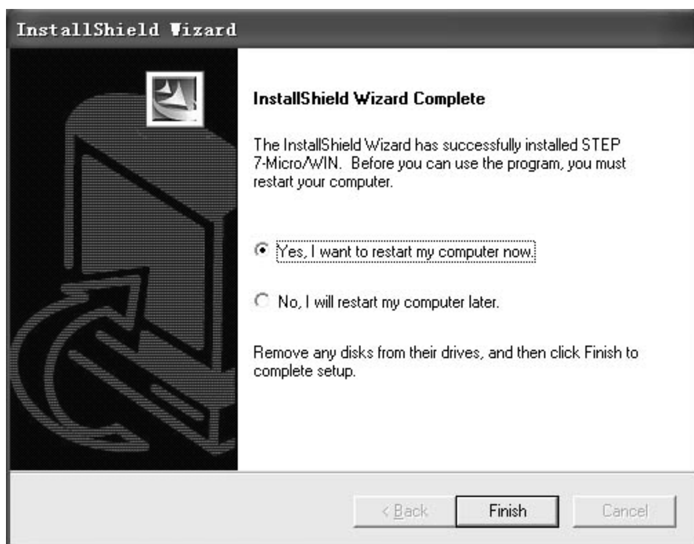


图 6-4　重新启动计算机

⑥ 在进入安装程序时曾选择英语作为安装过程中使用的语言。安装完成后，在用菜单命令“Tools”→“Options”打开的对话框的“General”选项卡中，选择“Chinese”。如图 6-8 所示，选择后，点“OK”，关闭界面再打开后就是中文的界面了。

3. Micro/WIN 指令库

STEP 7-Micro/WIN 还可以安装附加组件，如 Micro/WIN Instruction Library（指



图 6-5　选取并运行 STEP7-Micro/WIN 软件

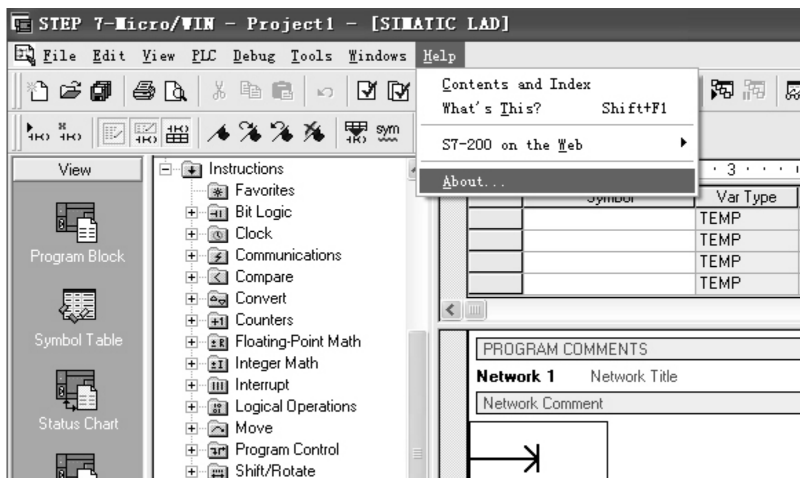


图 6-6　查看版本信息

令库）。

指令实际上就是编好的程序库，用户也可以定义自己的库程序，但西门子公司提供的 Micro/WIN 32 指令库，必须安装才能使用，西门子公司的指令库目前包括两个功能，即 USS 通信库和 MODBUS RTU 通信库程序。

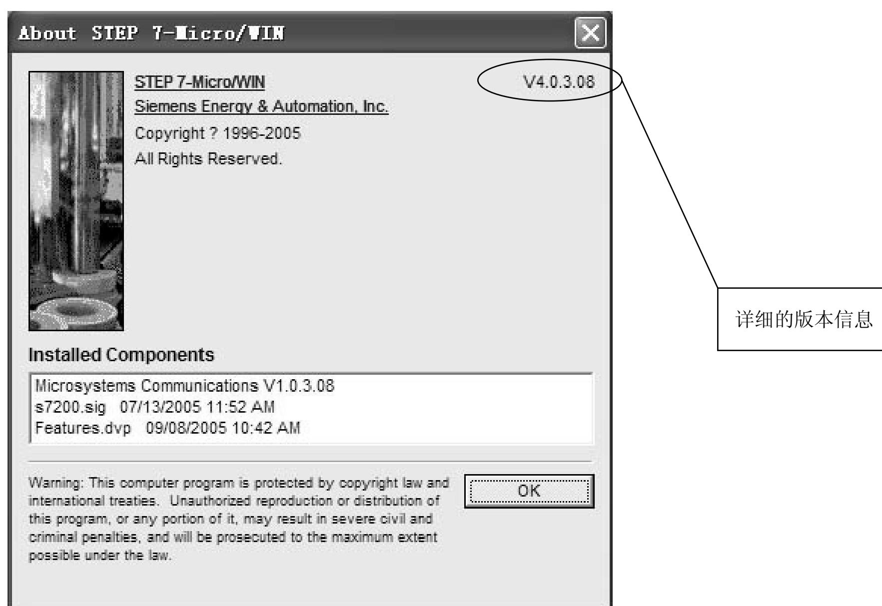


图 6-7　详细的版本信息

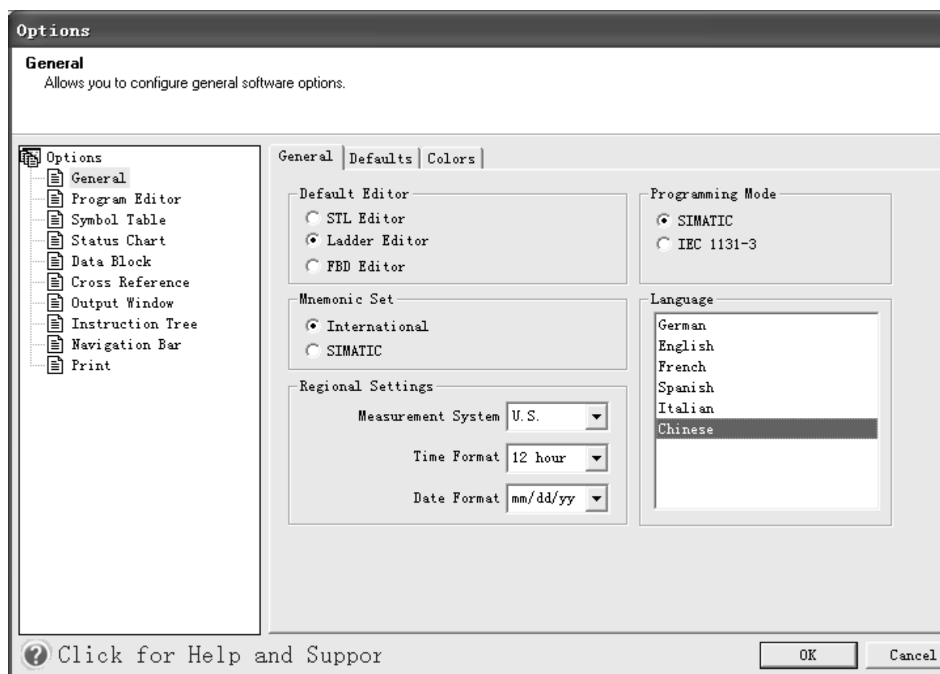


图 6-8　如何选择中文界面

如果计算机上已经安装了西门子公司提供的 Micro/WIN 32 指令库，安装新版本的 Micro/WIN 就会自动将库文件更新为最新版本；如果没有安装，则必须单独安装西门子公司的 Micro/WIN 32 指令库。安装指令库非常简单，只需要几秒钟就可以完成，像第五章中的 PLC 与变频器通信的例子，就要使用 USS 指令，而 USS 指令就必须通过指令库来调用，调用方法在第五章该例子中也说过了，此处不再赘述。

第二节 STEP 7-Micro/WIN 简介

1．STEP 7-Micro/WIN 窗口元素

STEP 7-Micro/WIN 的基本功能是协助用户完成应用软件的开发任务，例如创建用户程序，修改和编辑原有的用户程序；利用编程软件可设置 PLC 的工作模式和参数，编译、上载和下载用户程序，进行程序的运行监控等；它还具有简单语法的检查、对用户程序的文档管理和加密，以及提供在线帮助等功能。STEP 7-Micro/WIN 编程软件的主界面元素如图 6-9 所示。

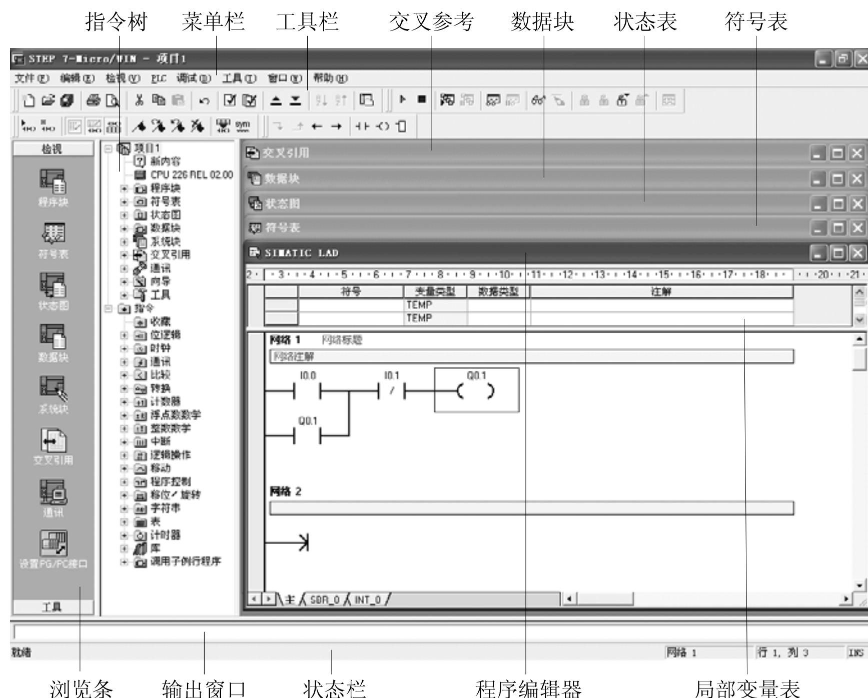


图 6-9 STEP7-Micro/WIN 主界面元素

主界面一般可分以下几个区：菜单栏、工具栏、浏览条、指令树窗口、输出

窗口、状态和程序编辑器、局部变量表（可同时或分别打开 5 个用户窗口）。除菜单栏外，用户可根据需要决定其他窗口的取舍和样式设置。

图 6-9 是 V4.0 版编程软件的界面，程序编辑就在此处进行。项目（Project）的名称由用户自己来定。下面介绍各部分的作用：

- 浏览条——显示常用编程按钮群组。浏览条包括两个组件框：检视和工具。

检视—显示程序块、符号块、状态表、数据块、系统块、交叉参考、通信和设置 PG/PC 接口 8 个按钮。

工具——显示指令向导、TD200 向导、位置控制向导、EM253 控制面板、扩展调制解调器向导、以太网向导、AS−i 向导、因特网向导、配方向导、数据记录向导和 PID 调节控制面板等十几个按钮。例如，我们想进行通信端口的参数设置，直接点击“系统块”，然后在弹出的对话栏中进行设置就可以了。

- 指令树——提供编程时用到的所有快捷操作命令和 PLC 指令。可以在项目分支里对所打开项目的所有包含对象进行操作，利用指令分支输入编程指令。可用“检视”菜单的“指令树”项决定是否将其打开。
- 交叉参考——查看程序的交叉引用和元件使用信息。
- 数据块——显示和编辑数据块内容。
- 状态表——允许将程序输入、输出或变量置入图表中，监视其状态。可以建立多个状态表，以便分组查看不同的变量。
- 符号表/全局变量表——允许分配和编辑全局符号。可以为一个项目建立多个符号表。
- 输出窗口——在编译程序或指令库时提供消息。当输出窗口列出程序错误时，双击错误信息，会自动在程序编辑器窗口中显示相应的程序网络。编辑好一段程序，如需检查是否有错误，可以直接点击“全部编译”按钮（它的图形符号就是两张白纸上面有一个蓝色的对钩），这时就会在输出窗口显示所编辑的程序是否有错、有几条错误，然后再点击“编译”按钮（它的图形符号就是一张白纸上面有一个蓝色的对钩），就会在此处显示错误在哪行哪列，如无错就显示“错误为 0”。
- 状态栏——提供在 STEP7-Micro/WIN 中操作时的操作状态信息。
- 程序编辑器——可用梯形图（LAD）、语句表（STL）或功能块图（FBD）编辑器编写用户程序，或在联机状态下从 PLC 上载用户程序，然后进行程序的编辑或修改。如果需要，可以拖动分割条以扩充程序视图，并覆盖局部变量表。单击程序编辑器窗口底部的标签，可以在主程序、子程序和中断服务程序之间移动。
- 局部变量表——每个程序块都对应一个局部变量表，在带参数的子程序调用中，参数的传递就是通过局部变量表进行的。

2. 菜单栏

允许使用鼠标单击或采用对应热键操作各种命令和工具，如图 6-10 所示。可

以定制“工具”菜单，在该菜单中增加自己的工具。菜单栏包含 8 个主菜单项。



图 6-10 菜单栏

菜单栏中各单项功能如下：

（1）文件（File）：文件操作可完成新建、打开、关闭、保存文件；.awl 文件的导入与导出；上载、下载程序和库操作；文件的页面设置、打印预览和操作等。

（2）编辑（Edit）：编辑功能完成剪切、复制、粘贴、选择程序块或数据块，插入、删除，同时提供查找、替换、光标定位等功能。

（3）检视（View）：选择不同语言的编程器（包括 LAD、STL、FBD 三种）；在组件中执行浏览条的任何项；可以设置软件开发环境的风格，如决定其他辅助窗口（浏览条窗口、指令树窗口、工具栏按钮区）的打开与关闭。

（4）PLC：可建立与 PLC 联机时的相关操作，改变 PLC 的工作方式（运行或停止）；在线或离线编译；清除程序和上电复位；查看 PLC 的信息和存储器卡操作、建立数据块、实时时钟、程序比较；PLC 类型选择及通信设置等。

（5）调试（debug）：主要用于联机调试，可进行扫描方式设置（首次或多次）；程序执行和状态监控选择；状态表的单次读取和全部写入；各种强制方式选择等。

（6）工具（tools）：可以调用复杂指令向导（包括 PID 指令、NETR/NETW 指令和 HSC 指令），使复杂指令的编程工作大大简化，安装 TD200 本文显示向导等；自定义界面风格（如设按钮及按钮样式，并可添加菜单项）；用“选项”子菜单也可以设置三种程序编辑器的风格，如语言模式、颜色、字体、指令盒的大小等。

（7）窗口（windows）：可以打开一个或多个窗口，并可进行窗口之间的切换，可以设置窗口的排放形式，如层叠、水平、垂直等。

（8）帮助（help）：通过帮助菜单上的目录和索引项可以查阅几乎所有相关的使用帮助信息；在软件编程操作过程中的任何步或任何位置都可以按 F1 键来显示在线帮助，或利用“这是什么”来打开相应的帮助，大大方便了用户的使用；帮助菜单还提供网上查询功能。

3. 工具栏

工具栏提供常用命令或工具的快捷按钮，通过简便的鼠标单击操作，就可完成相应的工作。如图 6-11 所示，可用“检视”菜单的“工具条”项目定义工具栏。其标准工具栏如图 6-12 所示。调试工具栏如图 6-13 所示。常用工具栏如图 6-14。LAD 指令工具栏如图 6-15 所示。

标准工具栏中，前面几种按钮与一般 Word 软件中的图形相同，作用也相同，后面的几种就是该软件特有的了。其中常用的有局部编译、全编译、上载、下载。

一个项目的程序编好后，用“全编译”及“局部编译”检查是否有错误；按“下载”键，将程序传入 PLC 中；按“上载”键，将程序从 PLC 中传入 STEP 7 中。



图 6-11　工具栏

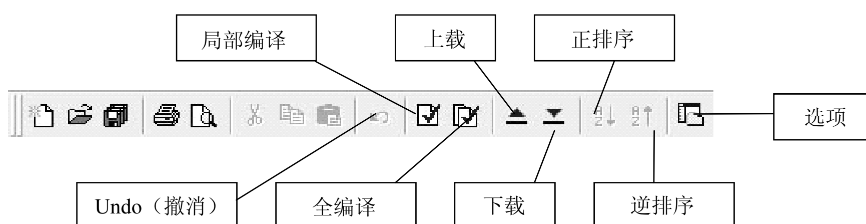


图 6-12　标准工具栏

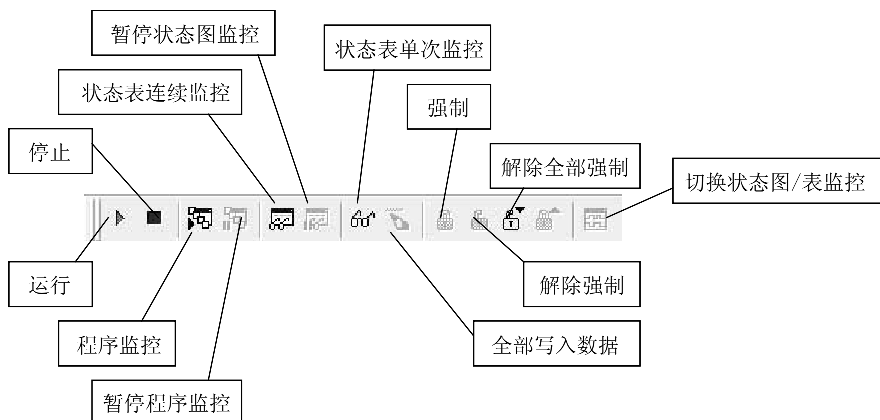


图 6-13　调试工具栏

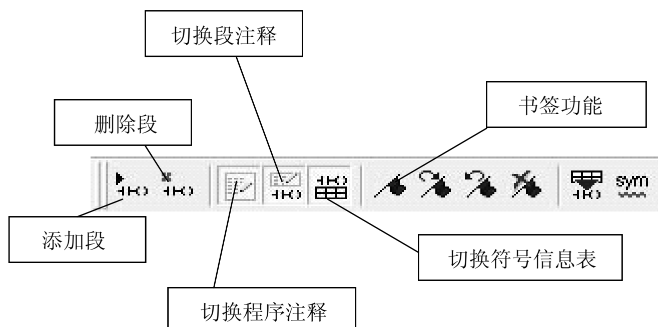


图 6-14　常用工具栏

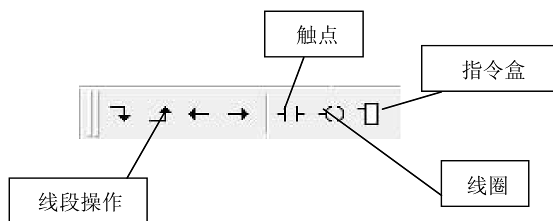


图 6-15 LAD 指令工具栏

4. 项目及其组件

STEP7-Micro/WIN 把每个实际的 S7-200 PLC 系统的用户程序、系统设置等保存在一个项目文件中，扩展名.mwp。打开一个 XXX.mwp 文件，就打开了相应的工程项目。

如图 6-16 所示，使用浏览条的视图部分和指令树的项目分支，可以查看项目的各个组件，并且在它们之间切换。用鼠标单击浏览条图标，或者双击指令树分支都可以快速达到相应的项目组件。

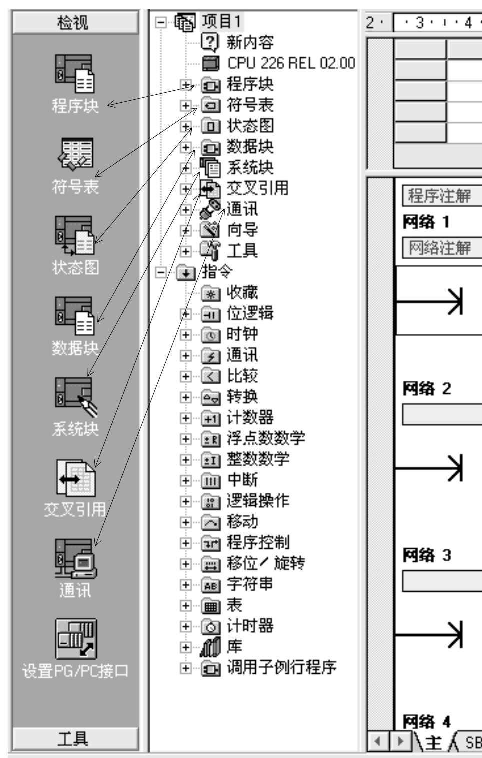


图 6-16 浏览条的视图部分和指令树的项目分支

例如，单击“通讯”图标可以寻找与编程计算机连接的 S7-200 PLC CPU，建立编程通信；单击“设置 PG/PC 接口”图标可以设置计算机与 S7-200 PLC 之间的通信硬件以及网络地址和速率等参数。

第三节　定制 STEP7-Micro/WIN

1. 显示和隐藏各种窗口组件

在菜单条中单击“检视”并选择一个对象，将其选择标记（一个对钩）在有和无之间切换。带选择标记的对象是当前在 STEP7-Micro/WIN 环境中打开的，如图 6-17 所示。



图 6-17　当前 STEP7-Micro/WIN 环境打开的对象

2. 选择“窗口”显示方式

在菜单条单击“窗口”，会出现如图 6-18 所示的选择项，当打开多个窗口时，用来决定窗口的排列方式，也可在不同窗口间切换，例如选择了“垂直”方式，窗口间的排列方式如图 6-19 所示。

3. 程序编辑的窗口选择

因为用户程序有主程序、子程序、中断程序之分，STEP7-Micro/WIN 为此也可以进行选择，也就是说各自都有自己的编程区域，如图 6-20 所示，只要用鼠标单击标签，即可在相应区域进行编程。

另外，用鼠标拖动分隔栏可以改变窗口区域的尺寸，如图 6-21 所示。

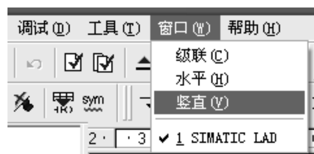


图 6-18　选择窗口显示方式

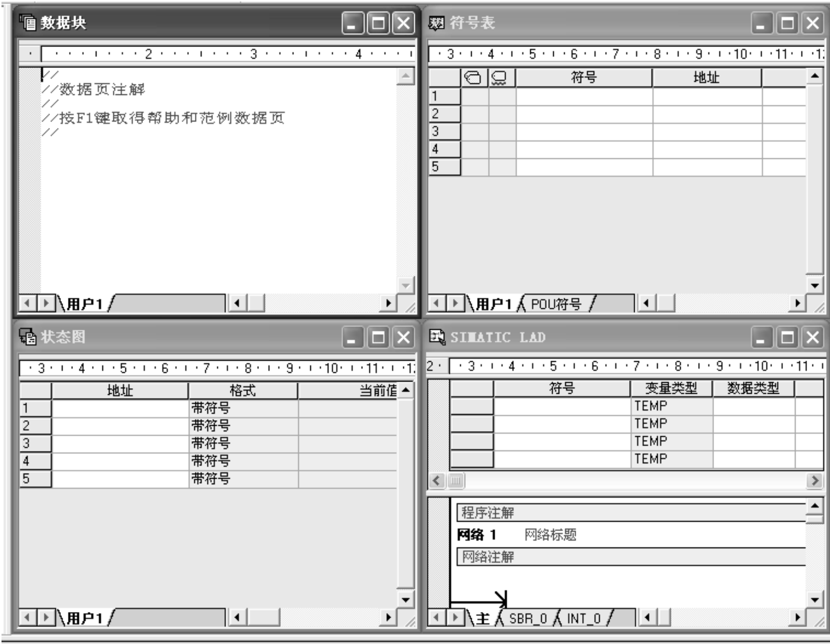


图 6-19 垂直方式窗口间的排列方式

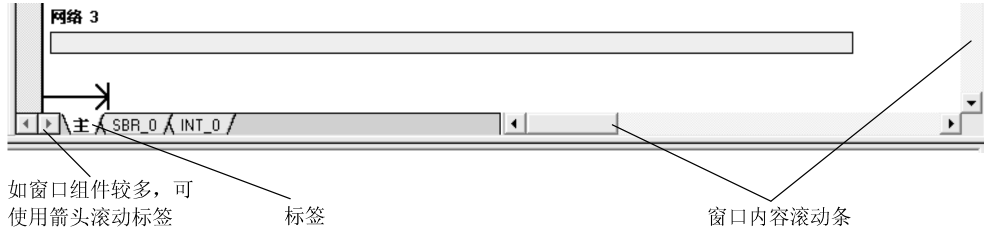


图 6-20 使用标签切换窗口的不同组件

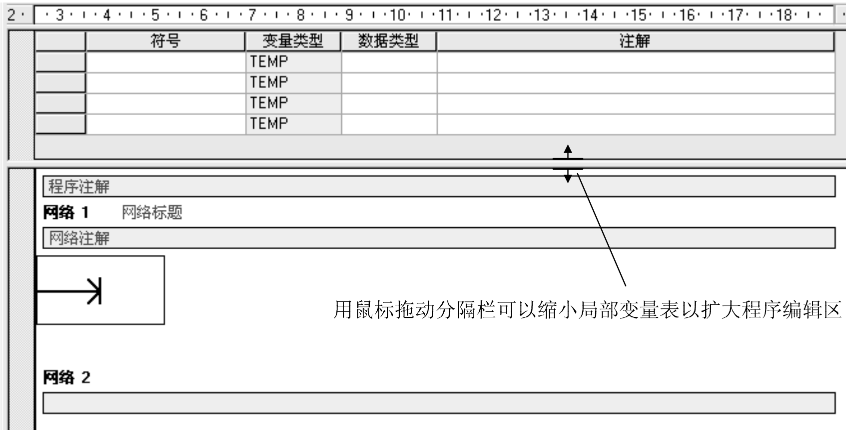


图 6-21 改变窗口区域的尺寸

第四节　编程计算机与 CPU 通信

在计算机上装好 STEP7-Micro/WIN 编程软件后，即可编写程序了，程序编好后要下载到 PLC 中，在下载之前还有一项工作要作，就是计算机与 PLC 要相互认识一下，这个过程被称为“通信”。相互认识后，才可以进行程序的上载或下载。

最简单的通信配置为：

（1）带串行通信端口（RS-232C 即 COM 口，或 USB 口）的个人计算机（PC），并已正确安装了 STEP7-Micro/WIN 的有效版本。

（2）PC/PPI 编程电缆，用此电缆连接计算机的 COM 口和 CPU 通信口；或者用 USB/PPI 电缆连接计算机的 USB 口和 CPU。

1. 设置通信

可以根据需要选择不同的通信波特率，9.6kbit/s 是 S7-200 PLC CPU 默认的通信速率。使用其他波特率需要在系统块内设置，并下载到 CPU 中才能生效。

用 PC/PPI 电缆连接 PG/PC 和 CPU，将 CPU 前盖内的模式选择开关设置为 STOP，给 CPU 上电。

① 用鼠标单击浏览条上的“通讯”图标出现通信窗口，如图 6-22 所示。



图 6-22　通信窗口

窗口左侧显示编程计算机将通过 PC/PPI 电缆尝试与 CPU 通信，右侧显示本地编程计算机的网络通信地址是 0，默认的远程（就是与计算机连接的）单台 PLC 的 CPU 端口地址为 2。

② 用鼠标双击右上角处的 PC/PPI 电缆的图标，出现如图 6-23 所示的窗口。单击“Properties...”（属性）按钮，查看或修改 PC/PPI 电缆连接参数及校准通信端口。

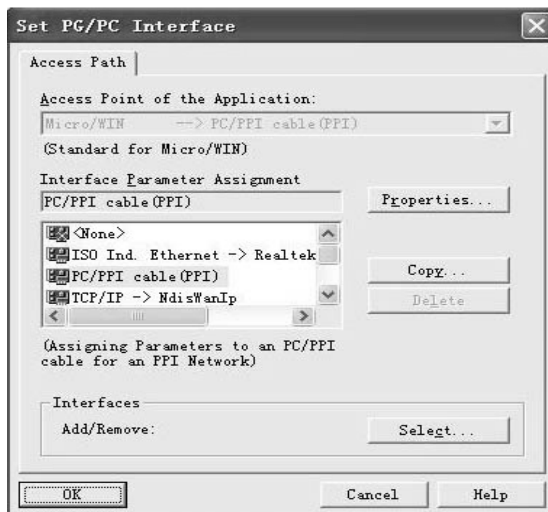


图 6-23　通信参数设置

③ 单击“Properties...”（属性）按钮后，出现两个选项，如图 6-24 所示，在“PPI”选项卡中查看、设置网络相关参数，在 Local Connection（本地连接）选项卡中，通过下拉选择框选择实际连接的编程计算机 COM 口（如果是 RS-232/PPI 电缆）或 USB 口（如果是 USB/PPI 电缆），如图 6-25 所示。这里选择的通信端口一定要与电缆实际连接的一致。

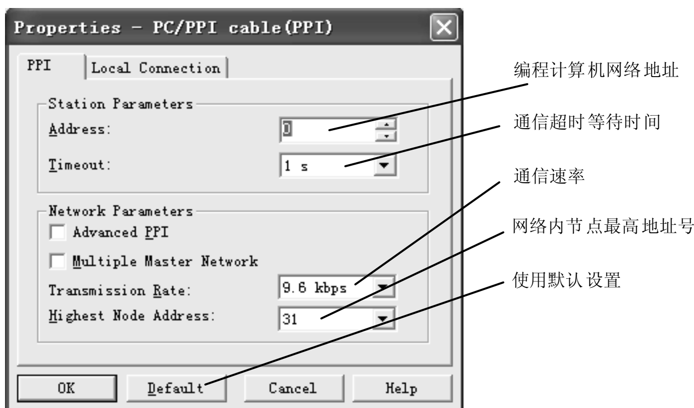


图 6-24　设置 PG/PC 界面

④ 单击“OK”按钮，回到“通讯”窗口，参见图 6-22，鼠标双击“双击以刷新”图标，开始进行编程计算机与 PLC 的通信（联络），也可以把它看成是编

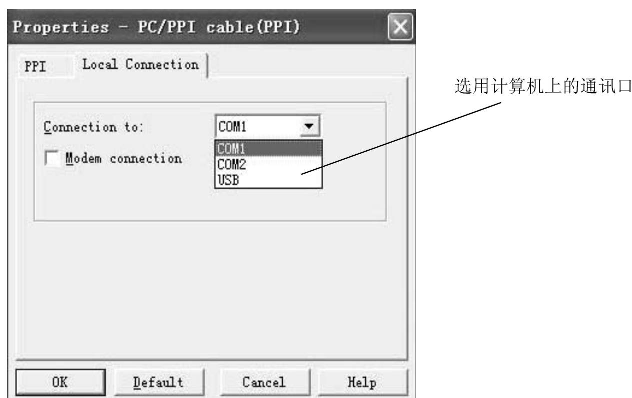


图 6-25　选择编程计算机通信口

程计算机搜索 PLC 的 CPU 信息，这一过程是完全自动进行的，如图 6-26 所示。



图 6-26　计算机正在与 PLC 通信

⑤ 在保证 COM 口（USB 口）设置准确、通信电缆完好无损的前提下，通信过程结束后，编程计算机肯定能搜索到与之连接的 PLC 的地址号、CPU 型号等，如图 6-27 所示。因为是单台 PLC，所以搜索到的默认地址应该是“2”。这时按“确认”键，通信过程就结束了。在这以后，就可以在计算机与 PLC 之间进行程序的上传或下载了。

2. PLC 信息

用鼠标双击找到的设备图标（这里是 CPU226 REL 02.00 地址：2），将显示 CPU 的信息，或在在线状态下执行主菜单命令“PLC”→“信息…”，将显示 PLC

的信息，如图 6-28 所示。包括操作模式、PLC 型号和 CPU 版本号、以 ms 为单位的扫描周期、I/O 模块配置、CPU 和 I/O 模块错误及历史事件日志等信息。

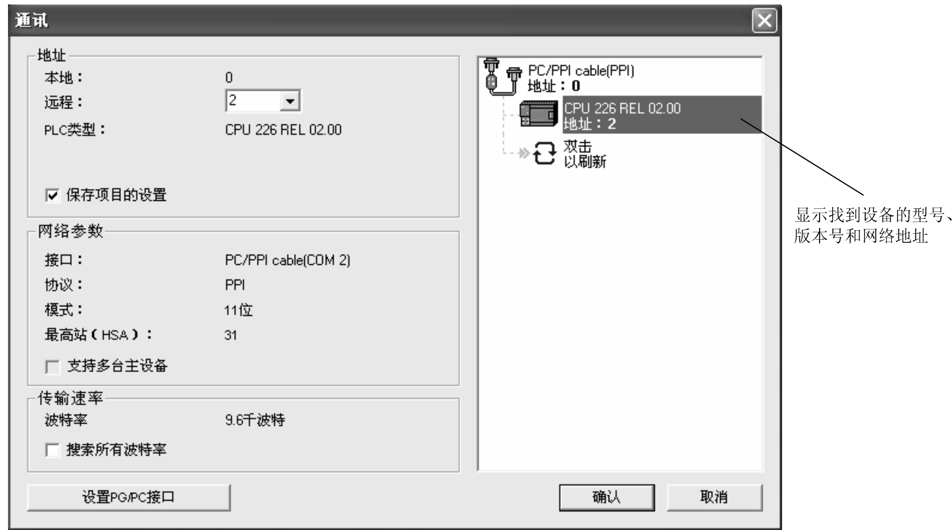


图 6-27 PLC 的信息搜索完毕



图 6-28 显示 CPU 信息

如果 PLC 类型支持“历史事件”日志，该按钮就会被使能。单击“历史事件...”按钮，查看何时上电、模式转换及致命错误的历史记录。只有设置了实时时钟，才能得到时间记录中正确的时间标记。

3. 读取远程 PLC 类型

双击指令树中 CPU 类型图标，或执行菜单栏命令“PLC”→“类型”，显示如图 6-29 所示的对话框；单击“读取 PLC”按钮，显示在线的 PLC 类型和 CPU 版本号；单击“确认”按钮关闭对话框后，发现指令树中的 PLC 类型处显示实际连接并通信成功的 CPU 型号和版本信息。也可在离线状态下，单击图中的列表框，在下拉选项中选择 PLC 类型或 CPU 版本。

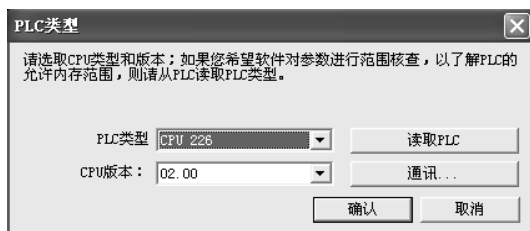


图 6-29　读取远程 PLC 类型

4. 设置实时时钟

要查看或设置存储在 PLC 中的当前时间和日期，选择菜单命令“PLC”→“实时时钟”，可以设置“PLC 时钟操作”对话框中的时间和日期，如图 6-30 所示。PLC 型号 214、215、216、221、222、224 和 226 支持“实时时钟”及 TODR/TODW 程序指令。CPU 222、224、226（2.0 版或更高版本）支持夏时制时间的自动调整。



图 6-30　设置实时时钟

第五节　程序的编写与传送

利用 STEP7-Micro/WIN 编程软件编辑和修改控制程序是用户要做的最基本工作，本节将以梯形图编辑器为例介绍一些基本编辑操作。语句表和功能块图编辑器的操作可采用类似的方法。

1. 项目文件管理

项目（project）文件来源有三个：新建一个项目、打开已有的项目和从 PLC 上载已有项目等。所谓项目，就是用户程序名称。

① 新建项目　在为一个控制系统编程之前，首先应创建一个项目。执行菜单命令“文件”→“新建”或按工具条最左边的【新建项目】按钮，可以生成一个新的项目。执行菜单命令“文件”→“另存为”可以修改项目的名称和项目文件所在的目录。

STEP 7-Micro/WIN 运行后，会在主窗口自动创建一个以“项目 1”命名的项目文件，主窗口会显示新建的项目文件主程序区。它是一组空的项目组件，包括程序编辑、数据块、符号表、交叉参考、状态表等 5 个用户窗口。STEP 7-Micro/WIN 支持 LAD（梯形图）、STL（语句表）和 FBD（功能块图）三种编程方式，图 6-31 为 LAD 程序编辑器窗口，是系统默认的编程方式。

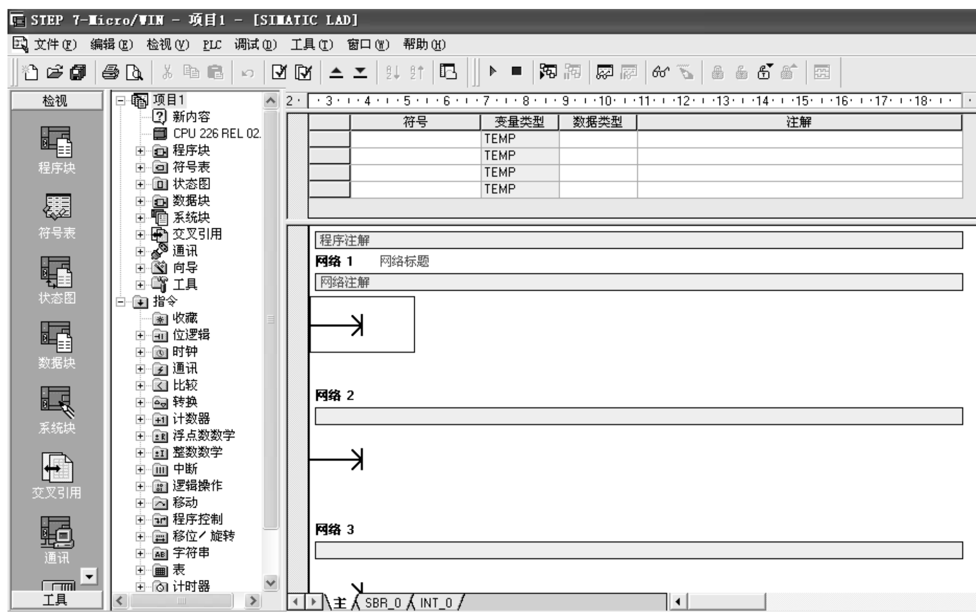


图 6-31　LAD 程序块窗口

程序块由主程序、可选的子程序（SBR_0）和中断程序（INT_0）组成。各程序分别包括程序注释、子程序注释、中断程序注释；程序段编辑区包括程序段网络编号、网络标题、网络注释和母线，单击“浏览条”中图标，直接切换项目的不同组件，如程序块窗口中用鼠标单击底部的程序标签可以在主程序、子程序和中断服务程序之间浏览。

② 打开项目　单击菜单栏“文件”中的“打开”项（参见图 6-10）或工具条中的“打开项目按钮”（位于图 6-12 的第二个），弹出“打开”对话框，选择项目路径及项目名称后，单击“确定”按钮，则打开现有项目。项目存放在扩展名为.mwp 的文件中。

也可选择在“文件”菜单底部所列出最近出现过的项目名称，直接选择打开。或者用 Windows 资源管理器找到要打开的项目，直接双击打开即可。

③ 上载项目　在确保计算机与 PLC 通信正常运行的前提下，如果要上载（PLC 至编辑器）一个 PLC 存储器中的项目文件（包括程序块、系统块、数据块），可用“文件”菜单中的“上载”项，也可单击工具条中的“上载”按钮（参见图 6-12 中的“▲”钮）来完成。上载时，选定要上载的块（程序块、数据块或系统块）、如图 6-32 所示，计算机会从 S7-200 PLC 的 RAM 中上载系统块，从 EEPROM 中上载程序块和数据块。上载来的程序一定要将名称选好后再保存，避免覆盖现象。



图 6-32　上载程序功能框

④ 项目保存和更名　如要在当前编辑操作状态下保存首次建立的项目文件，则单击菜单“文件”中“保存”或“另存为”命令，在工具栏中按“保存项目”按钮（图 6-12 中第三个）或按“Ctrl+S”组合键进行保存都可以。项目文件在以.mwp 为扩展名的单个文件中存储所有项目数据（程序、数据块、PLC 配置、符号表、状态表和注释）的当前状态，STEP 7-Micro/WIN 的默认文件名为“项目 1”，目录的默认值是之前确定的安装路径，当然可以根据自己的需要指定具体位置。

项目更名可使用菜单栏“文件”中“另存为”命令修改当前项目名称或目录位置；程序块中的主程序名（任何项目文件的主程序只有一个）、子程序名和中断程序名均可更改，方法是在指令树窗口中，右键单击需要更名的子程序或中断程序标签名，选中后直接键入所希望的名称。

⑤ 复制项目　使用编辑菜单命令或标准微软键组合方式，可实现项目段的全选（Ctrl+A）、复制（Ctrl+C）、剪切（Ctrl+X）及粘贴（Ctrl+V）等操作。其使用方法与普通文字处理软件相同。

⑥ 确定程序结构　较简单的数字量控制程序一般只有主程序（OB1），系统较大、功能复杂的程序除了主程序外，可能还有子程序、中断程序和数据块。

主程序在每个扫描周期被顺序执行一次。子程序的指令存放在独立的程序块中，仅在被别的程序调用时才执行。中断程序的指令也存放在独立的程序块中，用来处理预先规定的中断事件，在中断事件发生时将主程序暂时封存，由操作系统调用中断程序。

⑦ 添加子程序　如果在项目文件中有多个子程序，可以通过 3 种方法实现。

a. 在指令树窗口中，用鼠标右键单击“程序块”，在弹出的选择按钮中找到“插入”项，随后又出现两个可选项，一个是“子程序”，另一个是“中断”，单击“子程序”即可；

b. 用菜单命令“编辑→插入→子程序”来添加子程序；

c. 用鼠标右键单击编辑窗口（区域），在弹出的选项中选择“插入→子程序”。

新生成的子程序根据已有子程序的数目，自动递增编号（SBR_n）。

⑧ 添加中断服务程序　如果在项目文件中有多个中断服务程序，可以通过 3 种方法实现。

a. 在指令树窗口中，用鼠标右键单击“程序块”，在弹出的选择按钮中找到“插入”项，随后又出现两个可选项，一个是“子程序”，另一个是“中断”，单击“中断”即可；

b. 用菜单命令“编辑→插入→中断”来添加中断服务程序；

c. 用鼠标右键单击编辑窗口（区域），在弹出的选项中选择“插入→中断”。

新生成的中断服务程序根据已有中断服务程序的数目，自动递增编号（INT_n）。

2. 项目文件编辑

用选择的编程语言编写用户程序。梯形图直观方便、容易理解，一般都选择梯形图，梯形图程序被划分为若干个网络，一个网络中只能有一块独立电路，或者说一个网络中只允许一条支线与母线相连接，例如，图 4-14 仅需 1 个网络，而图 4-17 则需 5 个网络。一个网络中最多可写 32 行程序。如果一个网络中有两块或以上独立电路，在编译时将会显示“无效网络或网络太复杂无法编译”，程序“下载”就更谈不上了。

输入梯形图程序可以通过指令树、指令工具栏按钮、快捷键方式进行。程序块由可执行的指令代码和注释组成。

（1）输入编程元件　梯形图的编程指令（编程元件）主要有线圈、触点、指令盒（功能块）、标号及连线。输入编程指令的方法有以下几种：

1）在程序编辑区单击要放置编程指令的位置，此时会出现一个“选择方框”（矩形光标），然后在指令树所列的一系列指令中，双击要输入的指令符号，这个指令符号就自动落在矩形光标处，如图 6-33 中①所示。

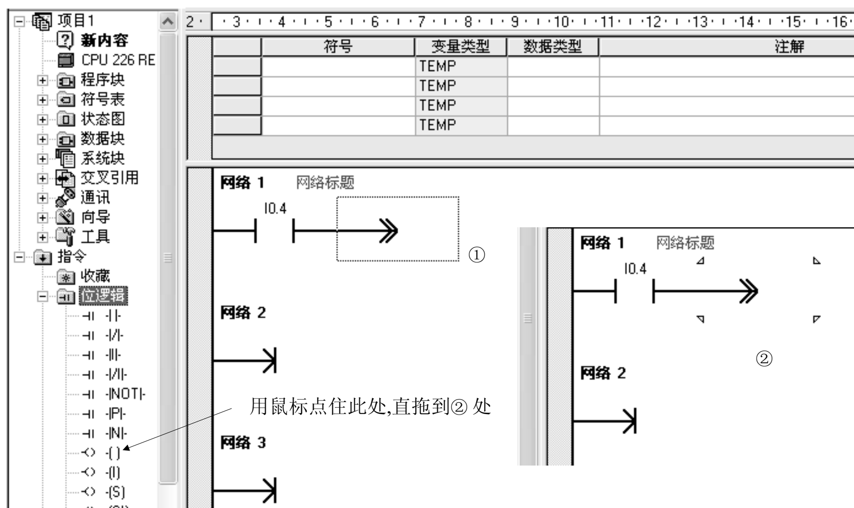


图 6-33　用鼠标输入编程元件

2）在指令树中单击所选择的指令并按住，将指令拖拽至程序编辑区需要放置指令的位置后释放鼠标按键，则相应指令就会落在该位置上，如图 6-33 中②所示。

3）在程序编辑区用鼠标确定指令所放置的位置，此时会出现一个“选择方框”（矩形光标），用工具栏上的一组编程按钮（参见图 6-15），单击触点、线圈或指令盒（功能块）按钮，或按对应的快捷键（F4=触点、F6=线圈、F9=功能块），从弹出的窗口下拉列表框所列出的指令中选择要输入的指令（利用鼠标拖动或键盘上的上下箭头键找到需用的指令），单击所需的指令或使用 Enter 键插入该指令，

如图 6-34 所示。

点击此处,会弹出下面的下拉列表

下拉列表

图 6-34　用快捷键输入编程元件

图 6-34 中的下拉列表是点击触点指令而产生的，指令段的终点处应该是线圈或者是指令盒（功能块），在工具栏上点击线圈按钮（图 6-15 中第 6 个）或者指令盒按钮（图 6-15 中第 7 个），同样会出现它们各自的下拉列表，也可以从指令树中点击“指令”获取所需要的线圈或功能块，如图 6-35 所示。放置方法与触点放置方法相同。

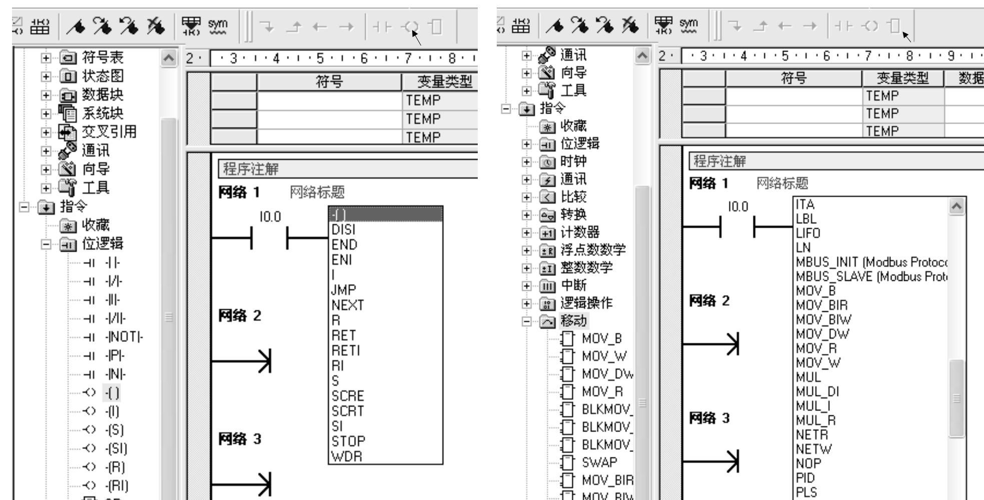


图 6-35　线圈、功能块下拉列表选择框

4）输入操作数。在用梯形图输入指令时，操作数最初是由红色问号代表，如图 6-36 中的“??.?”或“????”，表示参数未赋值。单击“??.?”或“????”处，或用光标（上、下、左、右箭头）键选择要键入操作数的指令后按 Enter 键，选择输入操作数的区域，选中后，此问号处就会被光标圈住，然后输入操作数即可。操作数输入完按 Enter 键，就会自动转入下一条指令的编辑。

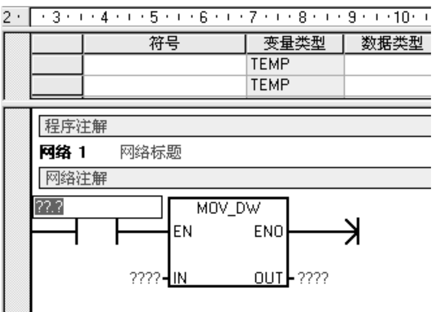


图 6-36　输入操作数

5）顺序输入和并联分支。顺序输入是从网络的母线开始连续在一行上依次输入各编程元件。编程元件是在矩形光标处被输入，编程元件以串联形式连接，输入和输出都无分叉。

并联分支是在同一网络块中第一行下方的编程区域单击鼠标，出现小矩形光标，然后输入编程元件生成新的一行，而且与上一行有连接关系。如输入与前边的程序无连接，出现同一网络有两条支线与母线连接，那就错了，则应在下一网络块中输入。

6）连接 LAD 线段。用工具条上水平和垂直线按钮（参见图 6-15），或按住键盘上的“Ctrl+光标”键，从光标位置处开始画线，连接编程元件以构成网络程序。例如，要在一行的某个元件后向上分支，可将光标移至要合并的触点处（见图 6-37），单击“上连线”按钮即可。如果要在一行的某个元件后向下分支，则将光标移到该元件，单击“下连线”按钮或用键盘操作完成连接，然后在进行其他编辑。

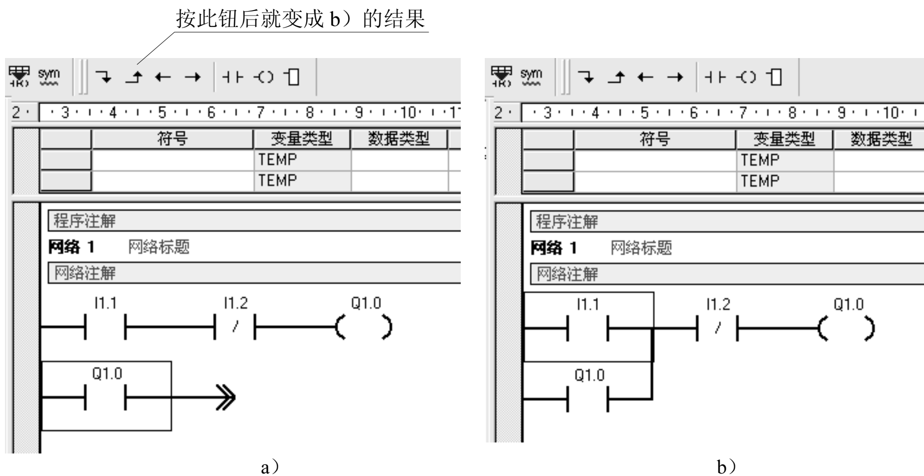


图 6-37　连线示意

STEP 7-Micro/WIN 支持与常用文档编辑软件类似的两种编辑模式：插入和改写。可用“Insert”键切换插入和覆盖两种编辑方式，在视窗状态栏右下角显示当前的 INS 或 OVR 模式状态。插入方式下，在一条指令上放新指令后，现有指令右移，为新指令让出位置；覆盖方式下，在一条指令上放新指令后，新指令替换现有指令。当用具有相同类型的方框覆盖（替换）一条指令时，对旧参数所作的任何赋值都保留到新参数。也就是说，如果第二个指令与第一个指令有同样数目的能流位输入、输入地址参数、能流位输出和输出地址参数，进行覆盖时参数赋值被保留。

7）编程语言切换。第四章与第五章中应用实例的程序，大多数是以语句表的形式给出的，这是为什么呢？回答很简单，是为了节省篇幅。梯形图被称为“电工图”，也就是说只要接触过继电接触器控制原理图的人，都能看得懂梯形图，梯形图直观明了，编写方便，特别适合编程调试阶段。在没有计算机软件编程之前，使用编程器给 PLC 输入程序，编程者既要用梯形图编写程序，又要会用语句表助记符输入程序，也就是说一定要掌握梯形图与语句表之间的指令对应关系才能完成整个设计工作。工作难度与强度都比现在大。

在 STEP 7-Micro/WIN 编程软件中可以将编写好的梯形图程序与语句表程序方便地进行切换。梯形图（LAD）、语句表（STL）或功能块图（FBD）三种编程语言表达模式用哪一种编写都可以，切换是通过工具栏中的“检视”来完成的，参见图 6-17，在检视的下拉菜单中的前三项就是 STL、LAD、FBD，需要使用哪一种点击它即可，只要在其前面出现“√”就是选中了。应该注意的是，在某一模式程序编好后，经编译不存在错误，方可进行切换，如有错误，则无法切换。

（2）输入注释 梯形图编辑器中共有 4 个注释级别，分别是项目组件注释、网络标题、网络注释和项目组件属性。在此可为每个 POU 及网络加标题或必要的注释说明，使程序清晰易读。

1）项目组件注释：单击“网络 1”上方的灰色文本框，键入 POU 注释，每条 POU 注释可允许使用的最大字符数为 4 096。POU 注释是供选用项目，反复单击“公用工具栏”中“切换 POU 注释”按钮或选择“检视”→“POU 注释”选项，可在 POU 注释“打开”（可见）或“关闭”（隐藏）之间切换。可视时，项目组件注释始终位于 POU 顶端，并在第一个网络之前显示。

2）网络标题：将光标放在网络标题行的任何位置，输入一个评价该逻辑网络功能的标题。网络标题中可允许使用的最大字符数为 127。

3）网络注释：单击“网络 n”（表示每个网络块或程序段）下方的灰色文本框，输入有关网络内容的说明，网络注释中允许使用的最大字符为 4 096。反复单击“切换网络注释”按钮或选择“检视”→“网络注释”选项，可在网络注释“打开”（可见）和“关闭”（隐藏）之间切换。

4）项目组件属性：右击程序编辑器窗口中的某一个 POU 标签，从弹出的快捷菜单中选择“属性”选项，打开“属性”对话框。“属性”对话框中有一般和保护两个标签，在“一般”标签中可依次设置名称、作者、程序编号等内容，在“保护”标签中可输入密码。

（3）编程元素的编辑　编程元素可以是单元、指令、地址及网络，编辑方法与普通的文字处理软件相似。在程序编辑器上选择要编辑的元素，通过工具栏按钮或“编辑”菜单命令，也可直接右击或使用快捷菜单选项，均可实现对选定对象的剪切、复制、粘贴、插入或删除等操作。

1）剪切、复制和粘贴。图 6-38 是在编程元件上右击鼠标时的结果，此时“剪切”和“复制”项处于有效状态，可以对元件进行剪切或复制。

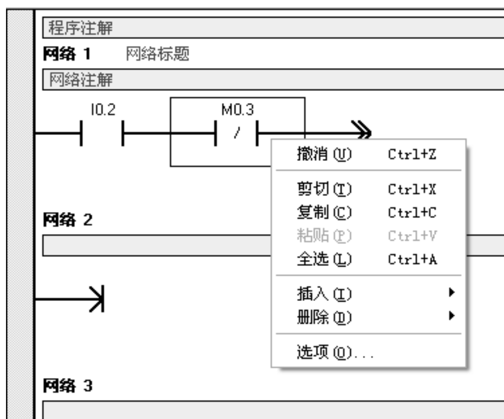


图 6-38　编程元件的编辑

用鼠标在梯形图母线上单击，可以选择该母线所对应的整个网络，如图 6-39 所示。在母线上按住左键拖动，可以选择多个网络段；也可先选择开始网络位置，然后在结束网络位置处按住 Shift 键并单击鼠标，确定多个网络段区域；可以在编辑器任意位置单击右键并通过下拉菜单完成“全选”操作（参见图 6-38）。选择后可进行剪切、复制和粘贴。粘贴操作只有在剪切、复制后有效。

2）插入和删除

● 编程时经常用到插入一行、一列、竖线、一个网络、一个子程序或中断程序等。

一行、一列、竖线、一个网络的插入方式是：在要插入处右击鼠标，弹出编辑菜单，选择“插入”，弹出下拉子菜单，如图 6-40 所示；单击要插入的项，然后进行编辑。也可用菜单栏“编辑”中相应的“插入”项来完成相同的操作。插入“行”或“列”是指在鼠标当前位置的上面或左边插入新的位置，“竖线”用来插入垂直的并联线段，“网络”是在光标上方插入网络并为所有网络重新编号。

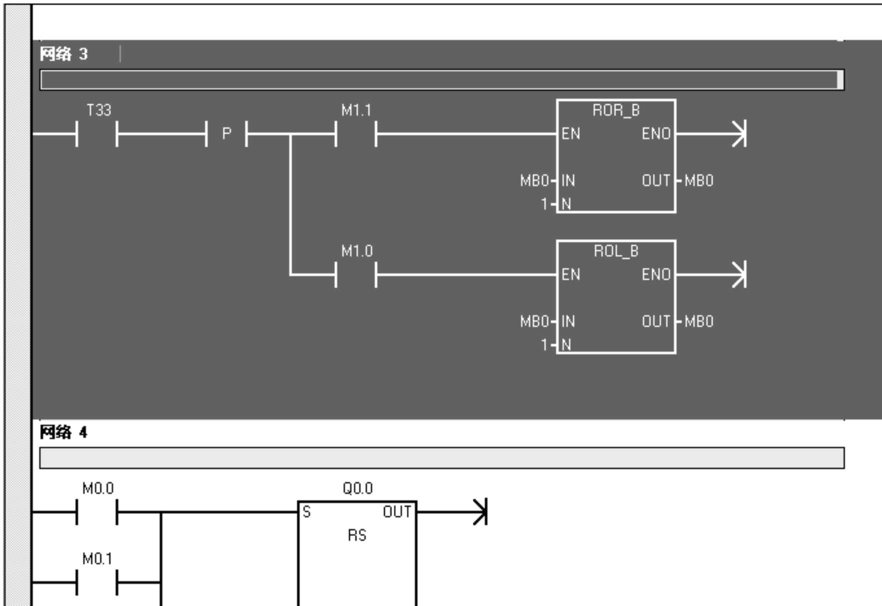


图 6-39 网络选择编辑

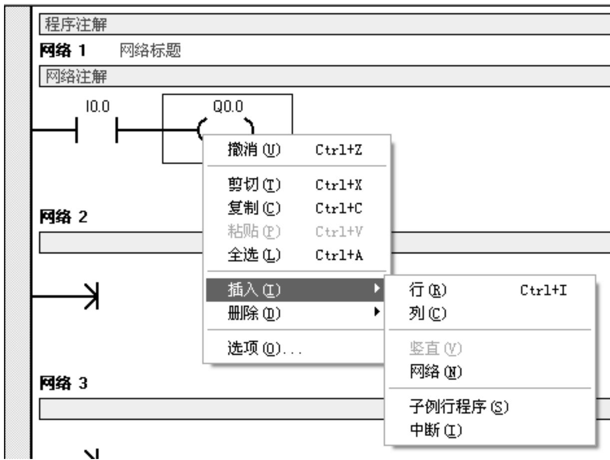


图 6-40 插入操作的选项菜单

● 编程时经常遇到删除一条指令、竖线、水平线段、一行、一列、一个网络、一个子程序或中断程序等操作。

一条指令、竖线、水平线段的删除方式是：单击要删除的指令、竖线的左侧位置、水平线段后按“Delete”键，删除相应的指令、竖线、水平线段。

一行、一列的删除方式是：在要删除行上的任意位置或要删除的某一列处单击鼠标右键，弹出快捷菜单，选择“删除”下拉子菜单中的“行”或“列”，删除

相应的行或列。

删除一个网络：在网络标题或网络注释上右击，选择“删除”下拉子菜单中的“网络”，删除相应的程序段。选择一个或多个程序段，按“Delete”键，或在被选择区域处右击，选择“删除”下拉菜单中的“选择”或“网络”，或单击“删除网络”按钮，删除程序中选择的整个网络。在下拉菜单出现后，按照快捷提示，用快捷键完成相应的操作。

删除一个子程序或中断程序：右击待删除的子程序或中断程序标签，选择“删除”下拉子菜单中的“POU”，弹出对话栏，问是否确定删除该项，点击“是”，或打开“指令树”中与之对应的文件夹，然后右击待删除的图标并选择弹出菜单中的“删除”命令，相应的子程序或中断程序被删除。

3）编译与下载。在 STEP 7-Micro/WIN 中，编辑的程序必须编译成 S7-200 PLC CPU 能识别的机器码，才能下载到 S7-200 PLC CPU 内运行。

程序编辑完成，可用菜单栏“PLC”→“编译”项或者用工具栏上的“编译”按钮（参见图 6-12），对当前编辑器中的程序进行离线编译。若选择“PLC”→“全部编译”，则按照顺序编译程序块（主程序、全部子程序、全部中断程序）、数据块、系统块等全部块，“全部编译”与哪一个窗口是否活动无关。

编译结束后在信息输出窗口显示编译结果。信息输出窗口会显示程序块和数据块的大小，也会显示编译中发现的语法错误的数量、各条错误的原因和错误在程序中的位置。双击信息输出窗口中的某一条错误信息，会在程序编辑器中相应出错位置出现矩形光标，如图 6-41 所示。必须改正程序中的所有错误，才能编译成功，进而进行“下载”操作。

上载和下载用户程序指的是用 STEP 7-Micro/WIN 编程软件进行编程时，PLC 主机和计算机之间的程序、数据和参数的传送。

下载之前，PLC 应处于 STOP 模式。单击工具条中的“停止”按钮，或选择“PLC”菜单命令中的“停止”项，进入 STOP 模式。如果不在 STOP 模式，可将 CPU 模块上的模式开关（处在 PLC 主机正面中右侧小门里的黄色开关）扳到 TERM 或 STOP 位置。

在计算机与 PLC 建立起通信连接后，如直接执行下载操作，STEP 7-Micro/WIN 会自动进行编译。用户程序编译成功后，可以将程序代码下载到 PLC 中去，而程序注释被忽略。

单击工具条中的“下载”按钮（参见图 6-12），或选择“文件”→“下载”项，将会出现下载对话框，如图 6-42 所示。用户可以分别选择是否下载程序块、数据块和系统块。单击“下载”按钮，开始下载信息，如 PLC 处于“RUN”模式，将出现“将 PLC 设置为 STOP 模式吗？”选项框，单击“是”按钮，使 PLC 转为 STOP 模式后，开始下载程序，同时输出窗口中显示“正在下载至 PLC…”信息，



图 6-41 在信息输出窗口显示编译结果



图 6-42 下载程序功能框

下载完毕后，显示“下载成功”字样。

如果 STEP 7-Micro/WIN 中设置的 CPU 型号与实际的型号不符，下载时会出现警告信息，这时应重新进行“通讯”并成功后再下载。

下载成功后，以手动方式将模式开关拨到“RUN”位置，或模式开关设为“TERM”位置，通过使用工具条中的按钮（参见图 6-13 中的第 1 个按钮“▼”），使 PLC 成为“RUN”模式。运行模式下，PLC 上黄灯“STOP”指示灯灭，绿灯“RUN”指示灯亮。

这时 PLC 就开始运行了，不管是否有输入信号，它都在周期性顺序扫描下载进去的程序。如果进行程序调试，就可以起动相应的输入信号开始调试了。

（4）数据块编辑　数据块用来对 V 存储器（变量存储器）进行数据初始化，可以用字节、字或双字赋值。数据块中的典型行包括起始地址以及一个或多个数据值，双斜线（“//”）之后的注释为可选项。键入一行后，按 Enter 键，数据块编辑器对输入行自动格式化（对齐地址列、数据、注释；大写 V 存储区地址标志）并重新显示。数据块编辑器接收大小写字母，并允许使用逗号、制表符或空格作为地址和数据值之间的分隔符。数据块的第一行必须包含明确的地址，以后的行可以不包含明确的地址。在单地址值后面键入多个数据或键入只包含数据的行时，由编辑器进行地址赋值。编辑器根据前面的地址和数据的长度（字节、字或双字）进行赋值。

选择菜单栏中的“检视”→“元件”→“数据块”，或者直接在“浏览栏”中单击“数据块”按钮，或者在“指令树”中单击“数据块”图标，均可打开数据块窗口进行操作。

（5）符号表　符号表是使用符号编址的一种工具表，可使程序逻辑更容易理解、便于记忆。使用符号表的方式有两种，一种是在编程时使用直接地址，然后打开符号表，编写与直接地址对应的符号名称，编译后由软件自动转换名称。另一种是在编程时直接使用符号名称，然后打开符号表，编写与符号名称对应的直接地址，编译后得到相同的结果。

要打开符号表，可单击“检视”菜单中的“符号表”项或浏览条窗口中的“符号表”按钮，在“符号”列键入符号名，使用 Tab、Enter 或 Arrow 键确认输入，同时移至下一个单元格。符号名最大允许长度为 23 个字符。在地址列和注释列分别键入地址和注释（注释为可选项，最多允许 79 个字符），符号表窗口如图 6-43 所示。右击单元格，可进行修改、删除、插入等操作。

一经编译，符号表就应用于程序中，图 6-44 显示编译程序后梯形图中的变量已经改为符号寻址的结果。

从图 6-44 中我们可以看出，在梯形图中用符号代替了地址，每个触点的作用比较明了，但还是觉得不太方便，希望能够同时看见符号和地址，这个要求也可以实现。

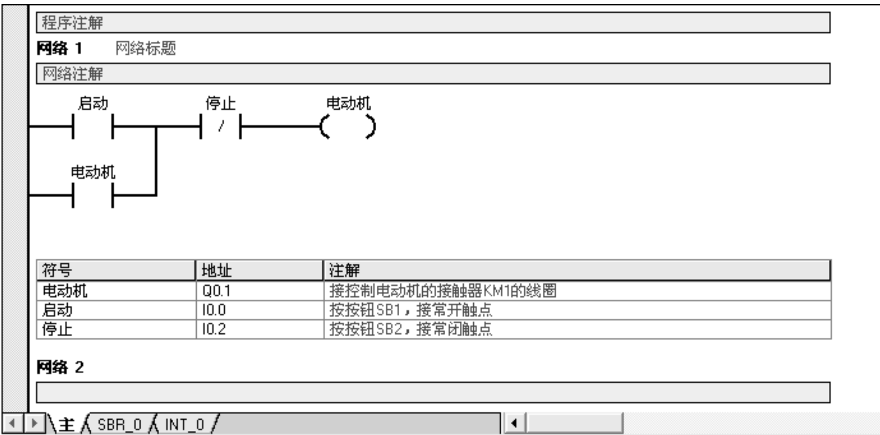
在“工具”菜单中单击“选项”按钮，在选项对话框中选择“程序编辑器”选项卡，在此对话框的右中间位置有个“符号编址”选项口，选择“显示符号和

地址”并确认，之后在所打开的项目程序元件上会显示符号和地址，如图 6-45 所示。

			符号	地址	注解
1			启动	I0.0	按按钮SB1，接常开触点
2			停止	I0.2	按按钮SB2，接常闭触点
3			电动机	Q0.1	接控制电动机的接触器KM1的线圈
4					
5					

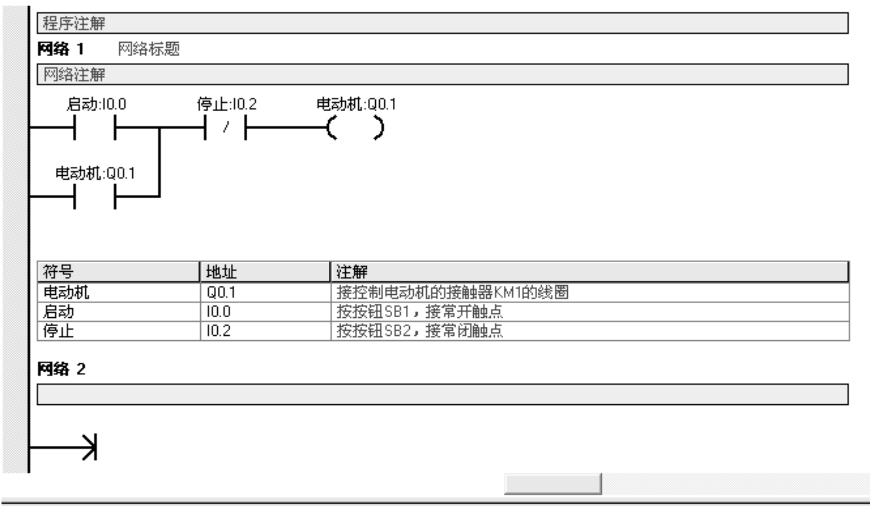
用户1　POU符号

图 6-43　符号表



符号	地址	注解
电动机	Q0.1	接控制电动机的接触器KM1的线圈
启动	I0.0	按按钮SB1，接常开触点
停止	I0.2	按按钮SB2，接常闭触点

图 6-44　使用符号表编程



符号	地址	注解
电动机	Q0.1	接控制电动机的接触器KM1的线圈
启动	I0.0	按按钮SB1，接常开触点
停止	I0.2	按按钮SB2，接常闭触点

图 6-45　同时查看符号和地址

第六节　程序的运行监控与调试

STEP 7-Micro/WIN 编程软件提供了一系列工具来调试并监控正在执行的用户程序。

1. 工作模式选择

S7-200 PLC 的 CPU 具有停止和运行两种操作模式。在停止模式下，可以创建、编辑程序，但不能执行程序；在运行模式下，PLC 读取输入，执行程序，写输出，反应通信请求，更新智能模块，进行内部事物管理及恢复中断条件，不仅可以执行程序，也可以创建、编辑及监控程序操作和数据。为调试提供帮助，加强了程序操作和确认编程的能力。

如果 PLC 上的模式开关处于“RUN”或“TERM”位置，可通过 STEP 7-Micro/WIN 软件执行菜单命令“PLC”→“运行”或“PLC”→“停止”进入相应工作模式。也可单击工具栏中的“运行”按钮（图 6-13 中第 1 个），或“停止”按钮（图 6-13 中第 2 个），进入相应工作模式，还可以手动改变位于 PLC 正面上小门内的状态开关改变工作模式，“运行”工作模式时，PLC 上的黄色“STOP”指示灯灭，绿色“RUN”指示灯亮。

2. 梯形图程序的状态监视

编程设备和 PLC 之间建立通信并向 PLC 下载程序后，STEP 7-Micro/WIN 可对当前程序进行在线调试。利用菜单栏中“调试”列表选择或单击“调试”工具条中的按钮，可以在梯形图程序编辑器窗口查看以图形形式表示的当前程序的运行状况，还可直接在程序指令上进行强制或取消强制数值等操作。

运行模式下，执行菜单命令“调试”→“开始程序状态监控”，或单击工具条中的“程序状态监控”按钮，用程序状态功能监视程序运行的情况，PLC 的当前数据值会显示在引用该数据的 LAD 旁边，LAD 以彩色显示活动能流分支。由于 PLC 与计算机之间有通信时间延迟，PLC 内所显示的操作数数值总在状态显示变化之前先发生变化。所以，用户在屏幕上观察到的程序监控状态并不是完全如实迅速变化的元件状态。屏幕刷新的速率取决于 PLC 与计算机的通信速率以及计算机的运行速度。

（1）执行状态监控方式　“使用执行状态”功能使监控视图能显示程序扫描周期内每条指令的操作数数值和能流状态。或者说，所显示的 PLC 中间数据值都是从一个程序扫描周期中采集的。

在程序状态监控操作之前选择菜单命令“调试”→“使用执行状态”（此命令行前面出现一个“√”即可），进入可监控状态。

在这种状态下，PLC 处于运行模式时，按下“程序状态”按钮（图 6-13 中第

3 个）起动程序状态监控，STEP 7-Micro/WIN 将用默认颜色（浅灰色）显示并更新梯形图中各元件的状态和变量数值，如图 6-46 所示。什么时候想退出监控，再按此按钮即可。

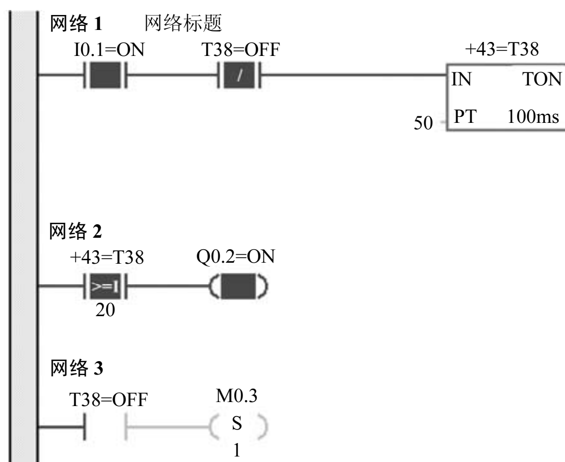


图 6-46　对 PLC 运行状态的监控

起动程序状态监控功能后，梯形图中左边的垂直“母线”和有能流流过的“导线”变为蓝色；如果位操作数为逻辑“真”，其触点和线圈也变成蓝色；有能流流入的指令盒的使能输入端变为蓝色；如该指令被成功执行，指令盒的方框也变为蓝色；定时器和计数器的方框为绿色时表示他们已处在工作状态；红色方框表示执行指令时出现了错误；灰色表示无能流、指令被跳过、未调用或 PLC 处于停止模式。

运行过程中，按下“暂停程序状态”按钮（图 6-13 中第 4 个），或者右击正处于程序监控状态的显示区，在弹出的菜单中选择“暂停程序状态（M）”，将使这一时刻的状态信息静止地保持在屏幕上以提供仔细分析与观察，直到再按一次“暂停程序状态”按钮（图 6-13 中第 4 个），才可以取消该功能，继续维持动态监控。

（2）扫描结束状态的状态监控方式　“扫描结束状态”显示在程序扫描周期结束时读取的状态结果。首先使菜单命令“调试”→“使用执行状态”命令行前面的“√”消失，进入扫描结束状态。由于快速的 PLC 扫描循环和相对慢速的 PLC 状态数据通信采集之间存在的速度差别，“扫描结束状态”显示的是多个扫描周期结束时采集的数据值。也就是说显示值并不是即时值。

在该状态 STEP 7-Micro/WIN 经过多个扫描周期采集状态值，然后刷新梯形图中各值的状态并显示。但是不显示 L 存储器或累加器的状态。在“扫描结束状态”下，“暂停程序状态”功能不起作用。

在运行模式下起动程序状态监控功能，电源“母线”或逻辑“真”的触点和线圈显示为蓝色，梯形图中所显示的操作数的值都是 PLC 在扫描周期完成时的结果。

3. 语句表程序的状态监视

语句表和梯形图的程序状态监视方法是完全相同的。用菜单命令“工具”→“选项”打开的窗口中，选择“程序编辑器”中的“STL 状态”选项卡，如图 6-47 所示，可以选择语句表程序状态监视的内容，每条指令最多可以监控 17 个操作数、逻辑堆栈中 4 个当前值和 11 个指令状态位。



图 6-47　语句表程序状态监视选择

状态信息从位于编辑窗口顶端的第一条 STL 语句开始显示。当向下滚动编辑窗口时，将从 CPU 获取新的信息。如果需要暂停刷新，还是按下“暂停程序状态”按钮（图 6-13 中第 4 个），过程与梯形图的相同。

4. 用状态表监视与调试程序

如果需要同时监视的变量不能在程序编辑器中同时显示，可以使用状态表监视功能。虽然梯形图状态监视的方法很直观，但受到屏幕的限制，只能显示很小一部分程序。利用 STEP 7-Micro/WIN 的状态表不仅能监视比较大的程序块或多个程序，而且可以编辑、读、写、强制和监视 PLC 的内部变量；还可使用诸如单次读取、全部写入、读取全部强制等功能，可以大大方便程序的调试。状态表始终显示“扫描结束状态”信息。

（1）打开和编辑状态表　在程序运行时，可以用状态表来读、写、强制和监视 PLC 的内部变量。单击浏览条中的“状态表”图标，或右击指令树中的“状态表”选项，在弹出的菜单中选择“打开”，或执行菜单命令“检视”→“元件”→“状态图”，均可以打开状态表，如图 6-48 所示。打开后可对它进行编辑。如果项目中有多个状态表，可以用状态表底部的选项卡切换。

	地址	格式	当前值	新数值
1	I0.1	位		
2	T38	位		
3	Q0.2	位		
4	M0.3	位		
5		带符号		
6		带符号		
7		带符号		
8		带符号		
9		带符号		
10		带符号		

用户1

图 6-48　状态表窗口

未起动状态表的监视功能时，可以在状态表中输入要监视的变量的地址和数据类型，定时器和计数器可以分别按位或按字监视。如果按位监视，显示的是它们的输出位的 ON/OFF 状态。如果按字监视，显示的是它们的当前值。

执行菜单命令“编辑”→“插入”，或用鼠标右键点击状态表中的单元，执行弹出的菜单中的“插入”命令，可以在状态表中当前光标位置的上部插入新的行。将光标置于最后一行中的任意单元后，按向下的箭头键，可以将新的行插在状态表的底部。在符号表中选择变量并将其复制在状态表中，可以加快创建状态表的速度。

（2）创建新的状态表　可以创建几个状态表，分别监视不同的元件组。用鼠标右键单击指令树中的状态图标或单击已经打开的状态表，将弹出一个窗口，在窗口中选择“插入”→“状态表”选项，可以创建新的状态表。

（3）起动和关闭状态表的监视功能　与 PLC 的通信连接成功后，用菜单命令“调试”→“开始图状态”或单击工具条上的“状态表”图标，可以起动状态表的监视功能，在状态表的“当前值”列将会出现从 PLC 中读取的动态数据，如图 6-49 所示。执行菜单命令“调试”→“停止图状态”或单击“状态表”图标，可以关闭状态表。状态表的监视功能被起动后，编程软件从 PLC 收集状态信息，并对表中的数据更新。这时还可以强制修改状态表中的变量，用二进制方式监视字节、字或双字，可以在一行中同时监视 8 点、16 点或 32 点位变量。

5. 在 RUN 模式下编辑用户程序

在 RUN（运行）模式下，不必转换到 STOP（停止）模式，便可以对程序作较小的改动，并将改动下载到 PLC 中。

状态表

	地址	格式	当前值	新值
1	I0.0	位	2#0	
2	I0.1	位	2#0	
3	Q0.0	位	2#1	
4	T38	位	2#1	
5	Q0.2	位	2#1	

用户定义1

图 6-49　状态表监控状态

建立好计算机与 PLC 之间的通信联系后，当 PLC 处于 RUN 模式时，执行菜单命令“调试”→“运行”中程序编辑，如果编程软件中打开的项目与 PLC 中的程序不同，将提示上载 PLC 中的程序。该功能只能编辑 PLC 中的已有程序。进入 RUN 模式编辑状态后，将会出现一个跟随鼠标移动的 PLC 图标。再次执行菜单命令“调试”→‘运行’中程序编辑，将退出 RUN 模式编辑。

编辑前应退出程序状态监视，修改程序后，需要将改动下载到 PLC。下载之前一定要仔细考虑可能对设备或操作人员造成的各种影响。

在 RUN 模式编辑状态下修改程序后，CPU 对修改的处理方法可以查阅系统手册。

6. 使用系统块设置 PLC 的参数

执行菜单命令“检视”→“元件”→“系统块”或直接单击浏览条中的“系统块”都可以打开系统块。单击指令树中“系统块”文件夹中的某一图标，则可以直接进入系统块中对应的对话框。

系统块主要包括：通信端口、断电数据保持、密码、数字量和模拟量输出表配置、数字量和模拟量输入滤波器、脉冲捕捉位和通信背景时间等，如图 6-50 所示。

打开系统块后，用鼠标单击感兴趣的图标，进入对应的选项卡后，可以进行有关的参数设置。有的选项卡中有【默认】按钮，按【默认】按钮可以自动设置编程软件推荐的设置值。

设置完成后，按【确认】按钮确认设置的参数，并自动退出系统块窗口。设置完所有的参数后，需要立即将新的设置下载到 PLC 中，参数便存储在 CPU 模块的存储器中。

7. 梯形图程序状态的强制功能

在 PLC 处于运行模式时执行强制状态。此时右击某元件地址位置，在弹出的菜单中可以对该元件执行写入、强制或取消强制的操作，如图 6-51 所示。强制和

取消强制功能不能用于 V、M、AI 和 AQ 的位。执行强制功能后，默认情况下 PLC 上的故障灯显示为黄色。

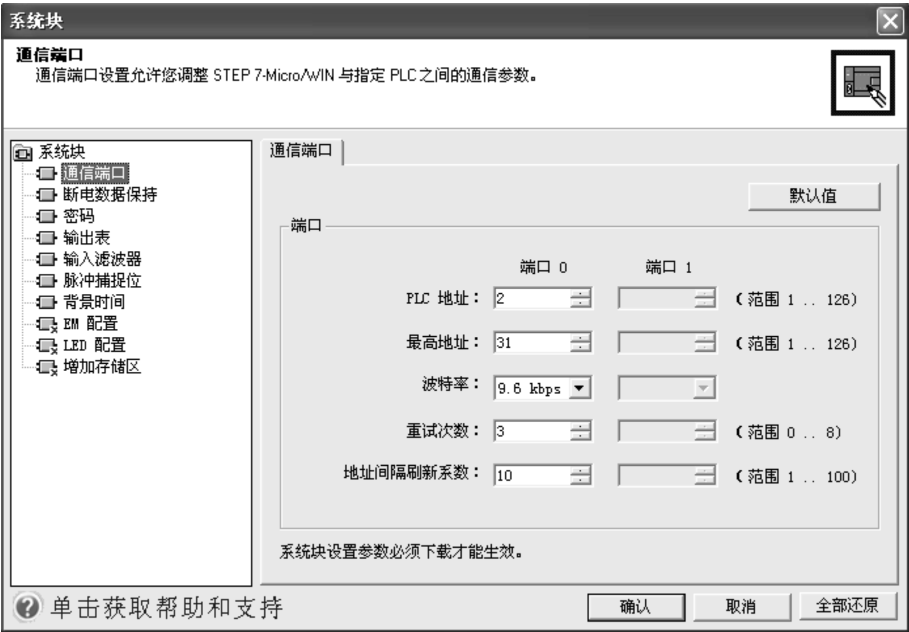


图 6-50 系统块选项框

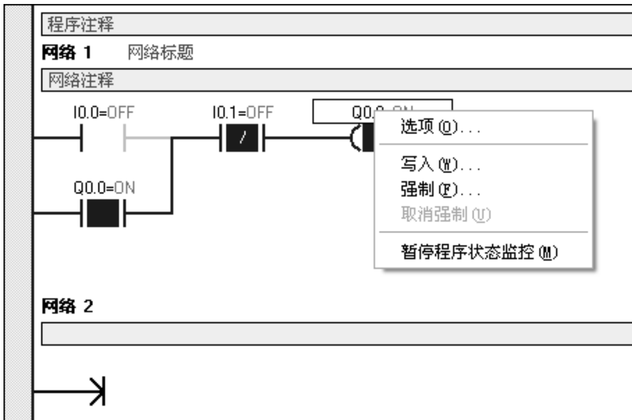


图 6-51 执行强制状态

在 PLC 处于停止模式时也会显示强制状态。但只有在非“使用执行状态”和“程序状态监控”条件下，起动菜单命令“调试”→“在停止模式中写入-强制输

出”后，才能执行对输出 Q 和 AQ 的写和强制操作。

8. 程序的打印输出

打印的相关功能在菜单栏“文件”项中，包括页面设置、打印预览和打印。

执行菜单命令“文件”→“页面设置…”，或单击工具栏“打印”按钮，在弹出的打印对话框中单击“页面设置…”按钮，出现页面设置对话框，如图 6-52 所示。



图 6-52　打印前的页面设置

可在页面设置对话框中单击“页眉/页脚…”按钮，弹出“页眉/脚注”对话框；可在该对话框中进行项目名、对象名称、日期、时间、页码以及左对齐、居中、右对齐的设定。

执行菜单命令“文件”→“打印预览”，或菜单工具栏“打印预览”按钮，显示打印预览窗口，可进行程序块、符号表、状态表、数据块、系统块、交叉引用的预览设置。如打印结果满意，可选择打印功能。

执行菜单命令“文件”→“打印”，或单击工具栏“打印”按钮，在如图 6-53 所示的打印对话框中，可选择需要打印的文件的组件的复选框，选择打印主程序网络 1～网络 20 的梯形图程序，但如果还希望打印程序的附加组件，例如，还要打印符号表等，则所选打印范围无效，将打印全部 LAD 网络。

单击“选项…”按钮，在出现的“打印选项”对话框中选择是否打印程序属性、局部变量表和数据块属性。



图 6-53 打印对话框

第七节 通信程序下载与向导编程

1. 主从式通信程序的下载

第四章的第十三题与第五章的第五题都是主从式通信的例子，那么程序编好后，如何通过 STEP 7-Micro/WIN 编程软件将程序下载到主站与从站中，继而投入运行呢？

这种通信方式接线很少，通过一根专用通信电缆将主站与从站通过指定的通信端口连接起来即可，通信端口在程序中应指定好，在这两例中都是用的“通信 1 口”。接下来是程序编辑，编程有两种途径：一是只在一台计算机上将主站程序编好后，下载到作为主站的 PLC 当中，然后编写从站程序，编好后下载到作为从站的 PLC 当中。二是用两台计算机分别给主站与从站编程下载，这一方法的优点是当两台 PLC 投入运行时，两台计算机可以分别监视主站与从站的工作状态。

计算机与 PLC 之间仍然用 PC/PPI 电缆进行通信，因“通信 1 口”已经被两台 PLC 之间通信电缆占用，所以计算机与 PLC 之间只好用“通信 0 口”。STEP 7-Micro/WIN 编程软件对单台 PLC 的默认站地址是“2”，就把主站地址设为 2，这样，主站的程序不难下载，因为只要计算机与 PLC 通信过，地址肯定是 2，编好程序直接下载就可以了。下载成功后将小门内的模式开关定在 RUN 位置，主站的编辑下载都结束了，就等待运行了。

将主站 PLC 的电源断开，然后拔下 PC/PPI 电缆插头，插到从站 PLC 的“通信 0 口”（如果是两台计算机分别编程下载就没有这一步），给从站 PLC 通上电源

并开始编程。

从站地址就应该从 3 开始，在这两个例子当中从站地址都选 3。将 PLC 站地址由 2 设置为 3，可点击“系统块”，选择“通信端口”选项卡（参见图 6-50），选项卡中第一项是“PLC 地址”的端口选项，将端口定为 3 后单击“确认”按钮退出。设置好的通信参数应立即下载到 PLC 主机，然后在进行通信时，会发现搜索来的站地址已经是 3。完成这几步后，再将已编写好的从站程序下载到作为从站的 PLC 当中，下载成功后将小门内的模式开关定在 RUN 位置，这时给两台 PLC 都通上电源，就可以调试运行了。

2. PLC 与变频器通信程序的下载

PLC 与变频器之间的通信使用的是 USS 通信协议，用户程序可以通过子程序调用的方式进行编程，编程的工作量很小。USS 协议指令在哪里找到，需要在 STEP 7 编程软件中先安装“STEP 7-Micro/WIN V32 指令库”，几秒钟即可安装好，USS 协议指令在此指令库的文件夹中。指令库提供 8 条指令来支持 USS 协议，如图 6-54 所示。

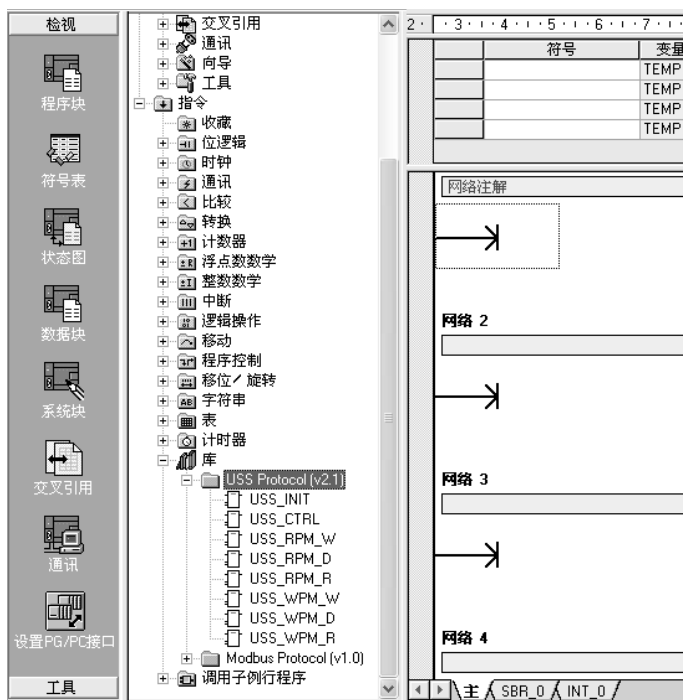


图 6-54　USS 通信协议指令库

调用一条 USS 指令时，将会自动增加一个或多个相关的子程序。调用方法是打开 STEP 7 编程软件，在“指令树”→“指令”→“库”→“USS Protocol”文件夹中，将会出现用于 USS 协议的通信指令，用它们来控制变频器和读写变频器

参数。用户不需要关注这些子程序的内部结构，只要将有关指令的外部参数设置好，直接在用户程序中调用它们即可，如图 6-55 所示。

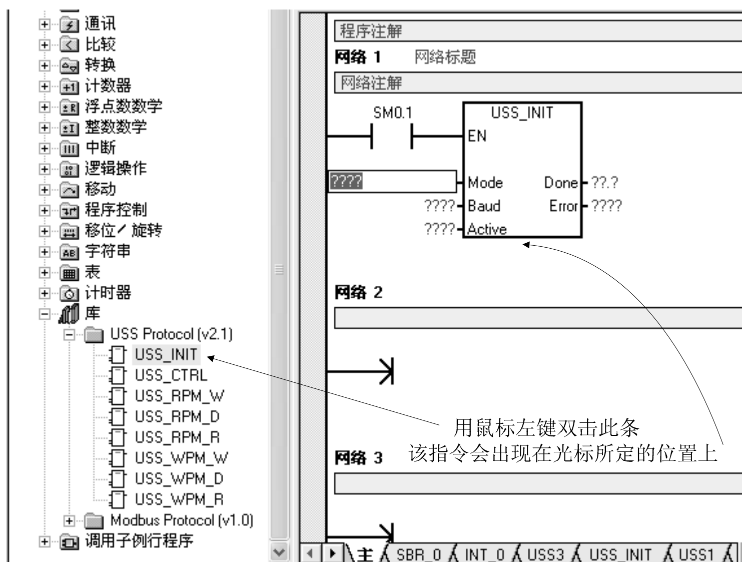


图 6-55 USS 通信指令的输入方法

- USS_INIT 指令用于初始化或改变 USS 的通信参数，只需在一个扫描周期调用一次就可以了，所以一般都使用 SM0.1 指令或在常开接点后加前沿微分指令达到只在一个扫描周期有效的目的。
- USS_DRV_CTRL（变频器控制）指令，在用户程序中，每一个被激活的变频器只能有一条。
- USS_RPM_x（读变频器参数）和 USS_WPM_x（写变频器参数）指令可以任意使用，但是每次只能激活其中的一条。

在下载程序调试之前，还应确保 PLC 与变频器之间的通信电缆已经接好，屏蔽线也已经接好，变频器操作面板上所设置的波特率和站地址等应与程序中的相符合。

3. 向导

一些特殊功能指令，像通信、高速计数器、PID、TD200 等，可以通过“向导”进行编程、设置，在 STEP 7-Micro/WIN 编程软件中，“向导”的功能相当强大，除了编程、设置之外，还可以利用它进行配方、数据归档、组态等。直接点击“工具”或者在“指令树”中点击“向导”，都会出现向导选项菜单，如图 6-56 所示。

例如，需要编辑高速计数器指令的应用程序，这时就可以使用“向导”，让“向导”为我们编程。只要按照对话框正确输入相关参数，“向导”编辑组态后，梯形

图程序就生成了。在图 6-56 中点击“工具”下拉选项菜单的第一项“指令向导…”，或在指令树的向导区域内双击“高速计数器”，都会出现“向导”编辑高速计数器程序的第一个对话框如图 6-57 所示。在对话窗口选定参数后，点击“下一步”按钮，继续进行后面的选择。

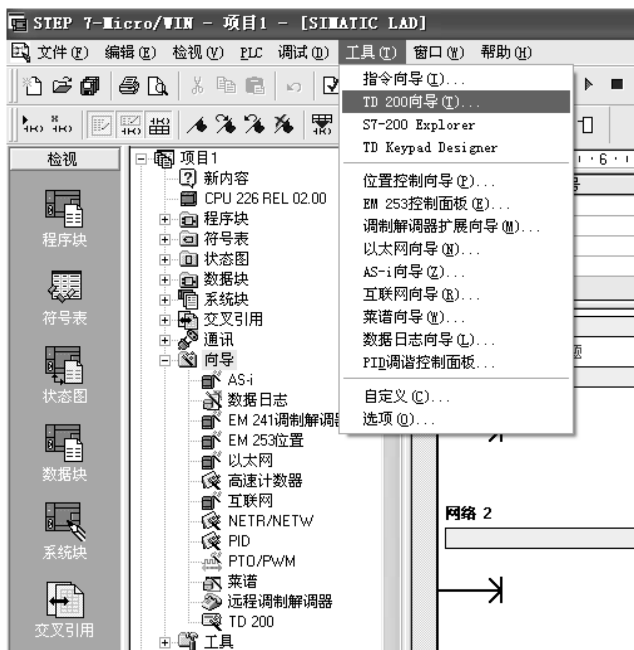


图 6-56 向导的选项菜单



图 6-57 “向导”编辑高速计数器程序的第一个对话框

参 考 文 献

［1］ 蔡行健，黄文珏，等. 深入浅出西门子 S7-200 PLC［M］. 北京：北京航空航天大学出版社，2003.

［2］ 王永华. 现代电器控制及 PLC 应用技巧［M］. 北京：北京航空航天大学出版社，2003.

［3］ 胡学林. 可编程控制器教程（基础篇）［M］. 北京：电子工业出版社，2003.

［4］ 廖常初. PLC 编程及应用［M］. 北京：机械工业出版社，2005.

［5］ 王兆义. 小型可编程控制器实用技术［M］. 北京: 机械工业出版社，2002.

［6］ 罗宇航. 流行PLC实用程序及设计［M］. 西安：西安电子科技大学出版社，2006.

［7］ 廖常初. PLC应用技术问答［M］. 北京：机械工业出版社，2006.

［8］ 李辉. S7-200PLC 编程原理与工程实训［M］. 北京：北京航空航天大学出版社，2008.

［9］ 吴中俊，等. 可编程序控制器原理及应用［M］. 北京：机械工业出版社，2005.

［10］ 张万忠. 可编程控制器入门与应用实例（西门子 S7-200 系列）［M］. 北京：中国电力出版社，2004.

［11］ 袁秀英. 组态控制技术［M］. 北京：电子工业出版社，2003.

ISBN 978-7-111-33698-3
策划编辑 ：牛新国
封面设计 ：路恩中

地址：北京市百万庄大街22号
电话服务
社服务中心：(010)88361066
销售一部：(010)68326294
销售二部：(010)88379649
读者购书热线：(010)88379203
邮政编码：100037
网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com
封面无防伪标均为盗版

上架指导：工业技术／电气工程／自动化



定价：48.00元